Trajectory Mapping Based-Object Tracking and CNN Based-Textural Feature Extraction for Application to Aquaculture

Trajectory Mapping Based-Object Tracking and CNN Based-Textural Feature Extraction for Application to Aquaculture

Muhamad Hilmil Muchtar Aditya Pradana

Supervisor: Prof. Keiichi Horio

Graduate School of Life Science and Systems Engineering
Kyushu Institute of Technology

This dissertation is submitted for the degree of
Doctor of Philosophy

September 2021

Publication

- **Accepted Paper**:

  1. H. Pradana, K. Horio., Tuna Nutriment Tracking using Trajectory Mapping in Application to Aquaculture Fish Tank. 2020 Digital Image Computing: Techniques and Applications, pp.1–8, 2020.
  2. H. Pradana, K. Horio., Automatic Controlling Fish Feeding Machine using Feature Extraction of Nutriment and Ripple Behavior. International Journal of Innovative Computing, Information and Control (IJICIC), 2021.

- **Submitting Paper**:

  1. H. Pradana, K. Horio., FB-IQA: No-Reference Image Quality Assessment using Different Convolutional Feature Sizes. The IEEE International Conference on Tools with Artificial Intelligence (ICTAI), 2021.

Acknowledgements

Pursuing this Ph.D.is a big life changing step for me. It would not be possible to achieve this without the support, guidance, and generosity that I received from many people.

I would like to first express my profound gratitude to my supervisor Prof. Keiichi Horio for his excellent guidance, patience, and kind suggestions for the past three years. I gratefully acknowledge the funding received towards my Ph.D. from Global AAR course where recommended me to be a candidate for the Japanese government Monbukagakusho (MEXT) scholarship.

A sincere thanks to all members of Horio Laboratory for giving me useful advice and their friendly support. I am especially grateful to Tashiro and Morio who were always helpful in numerous ways during my journey as an international student in Japan. Finally, I would like to say thank you to my best friends, my family for their continuous forceful and loving support, believing and understanding.

Abstract

Based on global aquaculture production statistic database, the proportion of aquatic animals farm is significantly increasing to 27.1 million tonnes in one decade. This trend indicates that aquaculture industry has to create a new technique to enlarge economic scale with reducing production cost and increasing production efficiency. Optimizing fish feeding process is the most influential aspect because the process itself takes up to 40 percent of total production cost. Automatic controlling fish feeding in real environment is still challenging problem and active research in aquaculture field because experienced fishermen can adequately control fish feeding machine based on assumption of ripple behavior and duration of fish feeding process. To build robust method which is reasonable application, we propose automatic controlling fish feeding machine based on computer vision using combination of two feature extractions which counts number of nutriments and estimates ripple behavior using regression and textural feature, respectively. To count number of nutriments, we apply object detection and tracking methods to acknowledge the nutriments moving to sea surface. Recently, object tracking is active research and challenging problem in applications of computer vision. Unfortunately, the robust tracking method for multiple small objects with dense and complex relationship is still challenging problem in aquaculture field with more appearance creatures. Assuming that degree of hunger can be represented by behavior of ripple area, estimation of ripple behavior is defined by human assumption of which the size and number of ripple can be used to adjust the activity level of ripple. Based on the number of nutriments fed and ripple behavior, we can control fish feeding machine which consistently performs well in real environment datasets. In evaluation, the proposed method presents the agreement for automatic controlling fish feeding by the activation graphs of regression and textural feature of ripple behavior results. Our tracking method can precisely track the nutriments in next frame comparing with other methods. Based on computational time, proposed method reaches 3.86 fps while other methods spend lower than 1.93 fps. Quantitative evaluation can give promising that the proposed method is valuable for aquaculture fish farm with widely applied to real environment datasets.

Keywords: Production Efficiency, Fish Feeding, Ripple Behavior, Textural Feature, Tracking, Multiple Small Objects

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos. From the perspective of engineering, it seeks to understand and automate tasks that the human visual system can do. Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information. Understanding in this context means the transformation of visual images into descriptions of the world that make sense to thought processes and can elicit appropriate action. This image understanding can be seen as the disentangling of symbolic information from image data using models constructed with the aid of geometry, physics, statistics, and learning theory.

The scientific discipline of computer vision is concerned with the theory behind artificial systems that extract information from images. The image data can take many forms, such as video sequences, views from multiple cameras, multi-dimensional data from a 3D scanner, or medical scanning device. The technological discipline of computer vision seeks to apply its theories and models to the construction of computer vision systems. One of application of computer vision is automatically optimizing fish feeding system.

Optimizing fish feeding process is the most influential aspect because the process itself takes up to 40 percent of total production cost [9, 11, 77]. Increasing company profit with controlling adjustment of nutriments is current problem of active aquaculture fish farm without wasting nutriments and dropping quality of fishes. Wasting nutriments create a big effect in water pollution because quality of water has a highly related to ensure survival rate of fishes and to enlarge fish fertility rate. Adjustment for giving nutriments is critical component to manage aquaculture development related to quality of water. From economic aspect, controlling fish feeding gives more benefits and determines quality of management expenses for aquaculture industry.

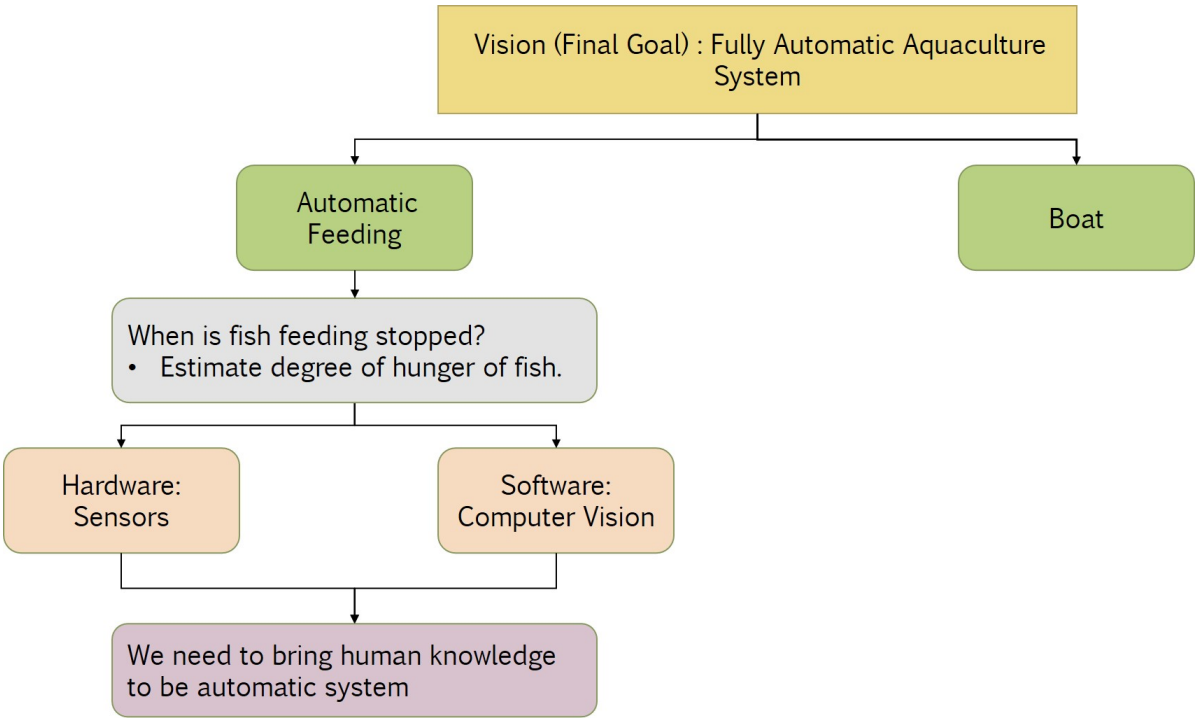


Fig. 1.1 Research map or vision of this research.

Final goal of this research is to create fully automatic aquaculture system which shown in Fig. 2.1. Firstly, building boat or ship is necessary to set up automatic fish feeding machine to do automatically. Next, automatic feeding itself is the most important aspect to build automatic system in aquaculture application. Automatic feeding is to define the optimal time to stop fish feeding machine. This automatic feeding is divided into hardware using sensor and software using computer vision. From both hardware and software, they need to imitate human knowledge to be automatic system.

In Fig. 2.2, we present the detail feature which can use to imitate human knowledge for automatic system. Sensor can be separated into attaching sensor in sea surface and in the fish body. On the other hand, using computer vision can use underwater camera and camera placed on boat or vessel. Video from camera placed on bot or vessel is more challenging because the video containing less information and only has nutriment given to the fish and ripple activity. By explaining of the problem, focusing of this thesis is a feature extraction from the information of nutriment given to the fish and ripple activity to estimate the activity of fish feeding.

The rest of this chapter is organized as follows: The explanation of fish feeding process has been introduced in Section 1.1 as well as their analysis by computer vision, including explanation detail of fish feeding processes, challenge and previous approaches. The aim and

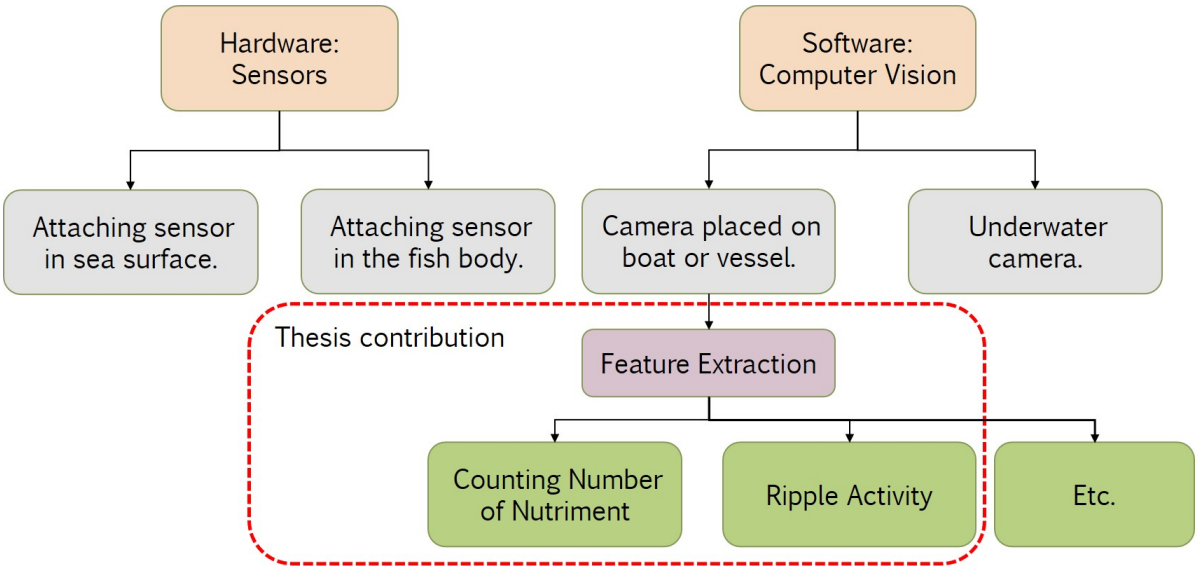


Fig. 1.2 How we estimate degree of hunger of fish and thesis contribution.

objective for this work are introduced in Section 1.2 including contribution of the proposed approach in Section 1.3. Finally, the outline of the thesis is described in Section 1.4.

1.1 Fish Feeding Process

Fish feeding in aqueous environment takes on dimensions beyond those considered in feeding land animals. These processes include the nutrient contribution of natural aquatic organisms in pond cultures, the effect of feeding on water quality, and the loss of nutrients if feeding is not consumed immediately. However, the concept of feeding is the same as that applied in feeding other food animals such as to nourish the animal to the desired level or form of productivity as profitably as possible. Thus, application of knowledge on the nutritional requirements of fish and the husbandry of feeding various cultured species is essential to successful aquaculture.

1.1.1 Nutrient Requirements

The nutrients required by fish for growth, reproduction, and other normal physiological functions are similar to those of land animals. They need protein, minerals, vitamins and growth factors, and energy sources. These nutrients may come from natural aquatic organisms or from prepared feeds. If fish are held in an artificial confinement where natural foods are absent, such as raceways, their feed must be nutritionally complete; however, where natural food is available and supplemental feeds are fed for additional growth, the feeds may not need to contain all of

the essential nutrients. Notable nutritional differences between fishes and farm animals are the following:

- Energy requirements are lower for fish than for warm-blooded animals, thus giving fish a higher dietary protein to energy ratio.
- Fish require some lipids that warmblooded animals do not, such as omega-3 series fatty acids for some species for crustaceans.
- The ability offish to absorb soluble minerals from the water minimizes the dietary need for some minerals.
- Fish have limited ability to synthesize ascorbic acid and must depend upon dietary sources.

Nutritional requirements of fish do not vary greatly among species. There are exceptions, such as differences in essential fatty acids and ability to assimilate carbohydrates, but these often can be identified with warm water or cold water and marine or freshwater species. The quantitative nutrient requirements that have been derived for several species have served adequately as a basis for estimating the nutrient needs of others. As more information becomes available on nutrient requirements of various species, the recommended nutrient allowances of diets for specific needs of individual species will become more refined.

1.1.2 Feeding Practices

Fish are fed in water. Feed that is not consumed within a reasonable time represents not only an economic loss, but can reduce water quality. Therefore, feed allowance, feeding method, and water stability of the feed are factors that the fish cultured must consider, but that the livestock feeder does not. The culture environment may make valuable nutrient contributions to the fish. For example, most waters contain enough dissolved calcium to provide most of the fish requirement. For fish that feed low on the food chain the pond environment can be a valuable source of protein, energy, and other nutrients.

1.1.3 Fish Feeding Efficiency

Fish convert practical feeds into body tissue more efficiently than do farm animals. The reason for the superior food conversion efficiency of fish is that they are able to assimilate diets with higher percentages of protein, apparently because of their lower dietary energy requirement. Fish, however, do not hold an advantage over chickens in protein conversion poultry convert

dietary protein to body protein at nearly the same rate as fish. The primary advantage of fish over land animals is lower energy cost of protein gain rather than superior food conversion efficiency. Protein gain per mega calorie of energy consumed is 47 for channel catfish versus 23 for the broiler chicken.

1.1.4 Challenge

There are several challenging of this research which shown in below:

1. The information of video contains less information and only has nutriment given to the fish and ripple activity.

2. The input video only uses one camera placed in above of vessel with a highly disturbance of ocean wave and many dense nutriments.

3. Collecting the dataset is very hard because the location of fish feeding farm is limited in Kyushu area and to make Memorandum of Understanding (MoU) with company takes time.

1.1.5 Previous Approaches

Recently, there are numbers of researchers creating several techniques to control amount of nutriments given to fishes. Those techniques can be defined as mechanical device controls [45, 102, 76, 7, 93, 107, 118, 97], and computer vision approaches [119, 79, 47, 121, 10, 83, 69, 120]. Mechanical device control uses external sensor which has different function for monitoring, identifying, and evaluating fish feeding behavior [86, 6, 8, 39]. External sensors can be categorized by acceleration sensors attached inside fish body and sensors related to water quality parameters. Accelerative data loggers is used to identify and to classify different kind of prey or nutriment types [45, 102, 76, 7]. In conclusion, this sensor has higher accuracy and lower false detection rate. However, it can create injury to fish body and installation sensor is costly for each fish body. Another type of safe sensor, without harming fish body [93, 107, 118, 97], is measurable sensors for detecting alteration of temperature and oxygen concentration. These sensors attach in underwater fish tank and calculate changing flow rate and temperature of the water caused by the activity of fish. Although these sensors create higher accuracy and prove save cost of fish feeding up to 29%, they can be easily interrupted by other parameters such as weather temperature.

On the other hand, computer vision approach has been commonly used in aquaculture industry because it performs in real-time controlling and low cost for maintainable equipment

[124, 41, 88]. This approach can be useful to classify gender and species, age and size measurement, quantity and quality inspection, counting and monitoring fish behaviors [119]. Based on wavelength signal imaging, it can be categorized into infrared [119, 79, 47, 121] and visible light [10, 83, 69, 120].The input video from infrared light is suitable to identify fish behavior and achieves better imaging effects under low visible light condition. However, it is suitable to identity fish behavior, analyzing of infrared imaging is remaining unsolved because object appearance from this imaging is unlike real condition. Beside that, application of detecting fish behavior using visible light gives several problems which only focuses on laboratory culture model[124, 41, 88]. It shows that the methods applied on laboratory culture model perform worst in real environment because several conditions cannot be anticipated such as flying bird, waste, appearance of fish tank boundary, and other undesirable objects. Automatic controlling fish feeding in real environment is still challenging problem and active research in aquaculture field because data appearance and weather condition can affect accuracy of detecting and tracking results. To build robust method, we propose automatic controlling fish feeding machine using combination of two feature extractions which estimate nutriment position and ripple behavior using regression and textural feature, respectively. Proposed method can ignore undesirable object by understanding behavior for each tracking object. Although proposed method cannot reach higher accuracy comparing use of sensors [45, 102, 76, 7, 93, 107, 118, 97], the advantage of our method can reduce equipment costs with requiring single camera placed above vessel. To analyze fish feeding behavior, tracking approach using regression and estimating ripple behavior using textural analysis are applied to make promising that the proposed method is valuable for aquaculture fish farm with widely applied to real environment datasets.

Object tracking is active research in applications to computer vision [40, 92, 82, 105, 74, 101, 117, 113, 12, 78]. Not only single-object tracking but also multiple-object tracking is unsolved problem since multiple-object association requires correctly collision in sequence frames. Multiple-object tracking is a more reliable solution rather than applying many of single-object tracking. The tracking problem can closely be determined to detection problem and even can be interpreted as an extension of object detection problem. Over the past decades, there are number of researchers focused on developing tracking-by-detection to generate more accurate and faster determination of object tracking [82, 105, 74, 101, 117, 113, 12, 78]. The tracking-by-detection method is the most popular tracking paradigm because object tracking is result of detection with same label in sequential frame. Good quality of the detection algorithms determines tracking result and these algorithms mostly use convolutional neural networks (CNNs) for the detection step. Proving that object tracking is an extension of object detection techniques is that the region proposal network (RPN), one of faster R-CNN detector

[85], adopts SiamRPN tracker [123, 59]. Multi-aspect-ratio anchors resolve bounding box estimation problem which has interference from previous trackers and significantly enhances the performance of siamese-network-based trackers. Unfortunately, the robust tracking method for multiple small objects with dense and complex relationship is still challenging problem in aquaculture field with more appearance creatures.

The idea of textural feature to understanding ripple behavior is applied from image quality assessment(IQA) index which can be represented as quantity of human perception referred to quality of image. Since texture is represented in photographic images, it is important to develop objective IQA index which has consistent value with perceptual similarity. In past decades, mean square error (MSE) and structural similarity (SSIM) had been the standard approach to evaluate the signal accuracy and quality although these approaches had the poor correlation result with human assumption and perception. Estimation of ripple behavior can also be defined by human assumption of which the size and number of ripple can be used to adjust the activity level of ripple.

1.2 Aim and Objective

The aim of this research is to resolve feature extraction from the information of nutriment given to the fish and ripple activity to estimate the activity of fish feeding. The main purpose of this research can be defined as follows:

- Tuna nutriment tracking based on trajectory mapping which can perform well comparing state-of-the-art multiple object tracking (MOT).
- A new novel small nutriment tracking method with collecting information of leading line into ripple.
- Significantly improvement result of trajectory mapping in real environment datasets.
- Ripple activity based on variance of VGG feature can perform well to separate between with and without ripple with simple and powerful method for understanding ripple behavior.
- Ripple activity based on variance of VGG feature is no longer need training and testing data.

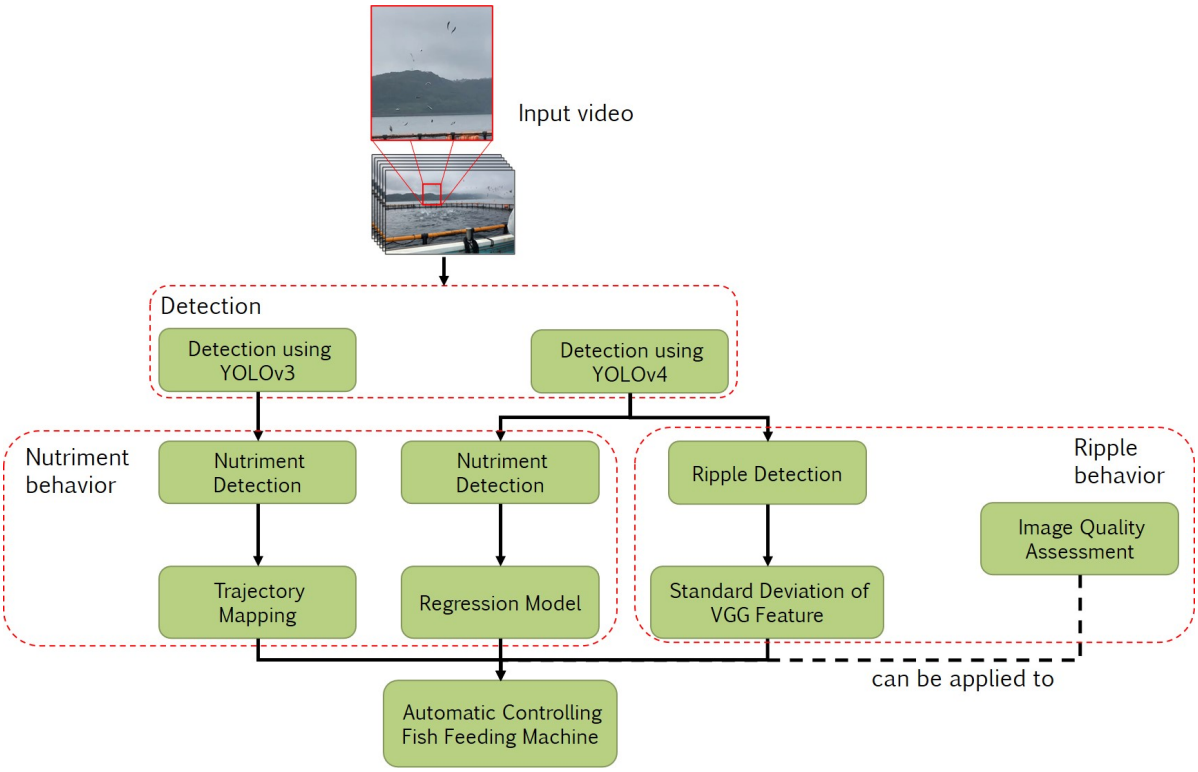


Fig. 1.3 Contribution of This Thesis

1.3 Contribution

The contribution of this thesis is to develop feature extraction from the information of nutriment given to the fish and ripple activity to estimate the activity of fish feeding which can be easier to identify degree of hunger of fish. The proposed method is able to extract feature extraction of nutriment and ripple behavior in real environment dataset. We also give suggestion method which is powerful method to identify ripple behavior using image quality assessment.

In Fig. 1.3, we show our contribution in this thesis. Firstly, we create of object detection model using two methods: YOLOv3 and YOLOv4. Beside that, to identify nutriment behavior is inputting nutriment detection and creates two methods: trajectory mapping and regression model. Trajectory mapping requires tuna nutriment detection using YOLOv3 while regression model uses YOLOv4 as input for nutriment detection. On the other hand, ripple behavior can be calculated using standard deviation of VGG feature of ripple area. We also present image quality assessment which trains using four different public databases. The idea of image quality assessment can bring another solution for computing ripple behavior by inputting ripple image and give an result of ripple activity.

1.4 Outline of The Thesis

In Chapter 2, the background and a comprehensive literature review are presented. The background will introduce conception of the method used for feature extraction in application to aquaculture problem. Regression analysis is the one of the method to solve tracking problem which is introduced in this chapter. Together with a comprehensive literature review of object detection and tracking. Image quality assessment is useful to understand ripple behavior also discussed in this chapter. Then the first approach of feature extraction of nutriment behavior is presented in Chapter 3 and the second approach is shown in Chapter 4. Trajectory mapping is the method which can handle faster object movement with projectile movement shown in Chapter 3. Then, we built improvement method using regression with simple combination of fully connected layers in Chapter 3. On the other hand, ripple behavior will be detailed in Chapter 5. In Chapter 5, variance of VGG feature is calculated to classify ripple area and non-ripple area. Chapter 6 will show image quality assessment model trained using other databases. Our model namely FB-IQA can burst learning accuracy without changing training parameter using adding some convolution layers. Finally, the conclusion of the whole thesis and the future work are given in Chapter 7.

Chapter 2

Literature Review

2.1 Regression Analysis

Regression analysis is a statistical technique for investigating and modeling the relationship between variables. Applications of regression are numerous and occur in almost every field, including engineering, the physical and chemical sciences, economics, management, life and biological sciences, and the social sciences. In fact, regression analysis may be the most widely used statistical technique.

As an example of a problem in which regression analysis may be helpful, suppose that an industrial engineer employed by a soft drink beverage bottler is analyzing the product delivery and service operations for vending machines. He suspects that the time required by a route deliveryman to load and service a machine is related to the number of cases of product delivered. The engineer visits 25 randomly chosen retail outlets having vending machines, and the in-outlet delivery time (in minutes) and the volume of product delivered (in cases) are observed for each.

2.2 Object Detection and Tracking

Detection can be viewed as a classification problem in which the task is to tell the presence or absence of a specific object in an image. If it is present, then the position of the object should be provided. Classification within a group of already detected objects is usually stated separately. However, in this case the question is formulated about what particular object is observed. Although the two groups are similar, recognition methods are left to the next chapter. Thus, examples of object detection in images are, for instance, detection of human faces, hand gestures, cars, and road signs in traffic scenes, or just ellipses in images. On the other hand, if

we were to spot a particular person or a road sign, etc. we would call this recognition. Since detection relies heavily on classification, as already mentioned, one of the methods discussed in the previous section can be used for this task. However, not least important is the proper selection of features that define an object. The main goal here is to choose features that are the most characteristic of a searched object or, in other words, that are highly discrimination, thus allowing an accurate response of a classifier. Finally, computational complexity of the methods is also essential due to the usually high dimensions of the feature and search spaces.

On the other hand, tracking of an object means finding the positions of this particular object in a sequence of images. In this process we take an indirect assumption that there is a correlation among subsequent images. Therefore for an image detected in one frame, it is highly probable that it will also appear in the next one, and so on. Obviously, its position and appearance can change from frame to frame. An object to be tracked is defined in the same way as for detection.

Recently, there has been increasing number of researchers focused on developing tracking algorithms to generate more accurate and faster determination of object. Tracking can be represented as a graph problem which can solved by a frame-by-frame [29, 19, 80, 103] or track-by-track [14, 116]. Interpretation of tracking problems with data association mostly uses a graph, where each detection is called as vertex, and each edge is pointing any possible link among them out as object tracked. Data association can be declared as minimum cost problem [15, 50, 62, 81] with learning cost problem [56] or motion pattern maps [58]. Alternative formulations to solve optimization problems is minimum clique problem [114] and lifted multicut problem [101] where its formulations follow body pose layout to obtain estimated model. Recently, efficient and robust tracking of multiple objects with complex relations remain unsolved. Hence, focusing to develop tracking algorithm in aquaculture is more challenging because tracked object has a lot of aquatic variant creatures.

2.3 Image Quality Assessment

According to necessity of reference images, objective IQA methods are separated into full-reference IQA (FR-IQA) [51], reduced-reference IQA (RR-IQA) [37], and no-reference IQA (NR-IQA) [112, 109]. FR-IQA is to compare quality of test image and reference image as standard of perfect quality image. Similar with FR-IQA, RR-IQA also uses reference image to estimate the quality of test image based on features extraction between them. Hence, NR-IQA has different techniques in which reference image is no longer used to estimate the quality of a test image that makes this method more challenging problem among them. Although FR-IQA and RR-IQA methods achieve better performance, availability of reference images

is limited in real application. So that, NR-IQA can give reliable solution for image quality in general application without requiring input of reference images. Meanwhile, the lacking of this method is that there are variant information poses and appearances which needs complex network structures.

In the beginning of developing NR-IQA methods, most of these are focused on artificial distortion type: blurring [60] and blocking feature appearance [61]. These approaches are only limited to recognize the type of distortion in the images. Lately, focusing on unknown distortions becomes more challenging problem to generalize an ability of NR-IQA methods [72, 111, 110, 38, 112, 109, 35, 87]. These methods propose the general characteristics of image distortions from feature learning [112] or image sketch [87] based on standardization of image quality models.

Recently, deep convolutional neural networks (DCNNs) give several contributions in computer vision applications [26, 27] such as image classification [27] and recognition [42], object detection [18] and tracking [26], natural language processing [36], and image quality assessment [122]. DCNNs have proven significantly better performance than the conventional methods because these networks consist of massive parameters, which can solve complex relationships. Several pretrained models of DCNNs in application to image classification are publicly available on internet such as AlexNet [54], VGG [90], Resnet [42], Inception [100], and Inception-Resnet [99]. These models were trained using ImageNet [25] and showed better performance to classify 1000 different classes. Widely application of DCNNs is to be feature extraction from images for further assessing image quality based on deep learning approach.

2.4 Dataset

2.4.1 Dataset Information

Our datasets contain a video of fish feeding process which has small and dense nutriments with 418 frames in which the detail information is shown in Fig. 2.1. Each dimension of video frame is 1920 $\times$ 1080 pixels. This sequential video is saved in MOV format with frame rate 30 frames/second. Range size of nutriment is starting from 9 $\times$ 6 to 13 $\times$ 36 pixels.

2.4.2 Dataset Collection

Fish farm is available in limited location in Japan. One of fish farm location is in Oita prefecture. So we use boat to go there to collect the dataset in which the location of fish farm is in the sea place shown in Fig. 2.2.

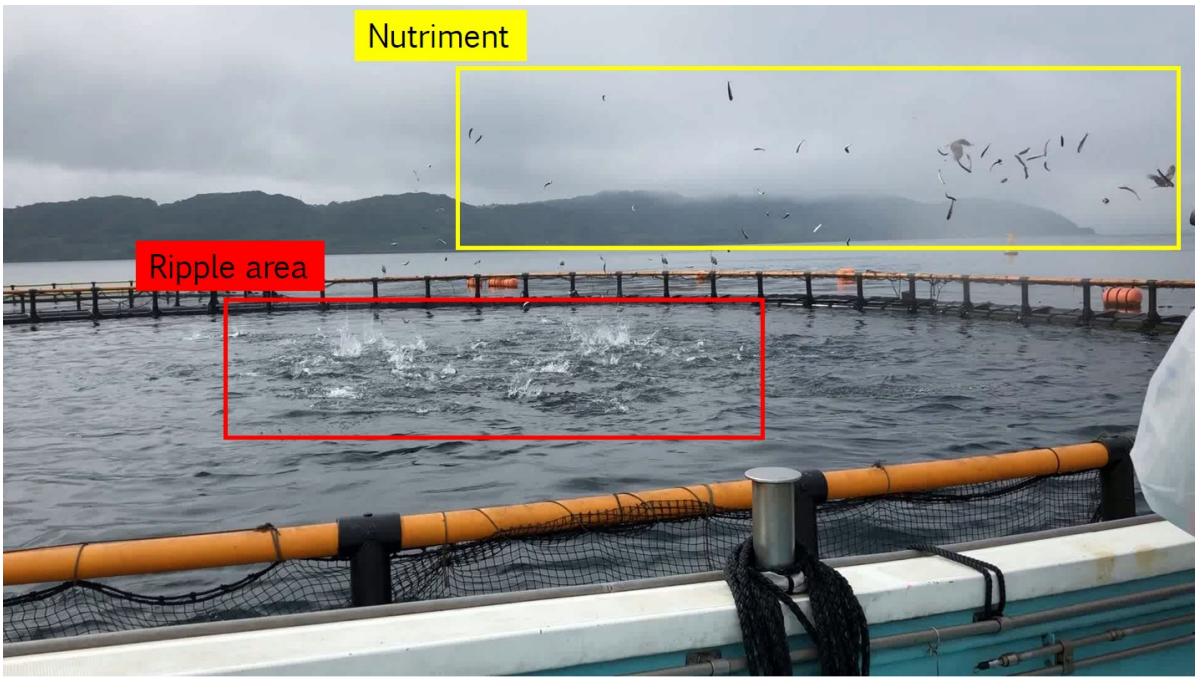


Fig. 2.1 Detail of the video used for this research.

Fig. 2.2 Data collection in Oita Prefecture.

Table 2.1 Hardware and software environment for running proposed method and state-of-the-art benchmark methods for comparison.

		Spesification
Hardware	CPU	Intel Core i7-9700 CPU @3.00GHz (8 CPUs)
	RAM	16 GB
	GPU	NVIDIA GeForce GTX 745
Software	OS	Windows 10 Pro 64-bit
	IDE	Microsoft Visual Studio Professional 2017 v.15.9.25
	Language	Python 3.6.6 64bit

2.5 Hardware and Software

To describe the computational complexity and execution time of the proposed methodology, a computational time analysis is conducted using a video with 418 frames. The specification of hardware and software used to analyze proposed method and state-of-the-art benchmark methods is shown in Table 2.1.

Chapter 3

Trajectory Mapping

3.1 Introduction

Aquaculture is the one of farming type in which aquatic creatures require acceptable environment for living habitat and availability nutriment to increase productivity and sustain healthy growth [32, 33, 24, 67]. Within current requiring acceptable habitat, water quality is also a vital component to enlarge fish fertility rate [24, 67, 31, 65]. Water quality can be obtained by cleaned often and give optimal amount of nutriment. Increasing number of nutriment can affect a lot of foods wasted in the water and quality of water occurs highly polluted. On the other hand, reducing feeding will lead starvation and drop fish quality. So that, management of nutriment delivered is vital component to balance productivity rate [43, 13].

The cost of fish feeding is usually around 40 percent of total production cost [9, 11, 77]. Estimating a state of fishes in a tank and adjusting an amount of nutriments play an important role to manage cost of fish feeding system. It is applied to control the amount of nutriment and realizes the fish behavior in tank. Lately, application to monitor fish behavior has been adopted by a telemetry-based approach [20, 22] and a computer vision(CV)-based approach [23, 108, 96, 125, 28, 10, 3].

A telemetry-based approach is a technique attaching an external transmitter by mounting, or surgical implantation in the peritoneal cavity [20]. Attaching a transmitter in each fish will spend higher cost and its transmitter can only set in large fish. When their fishes had been farmed, attachment will always be given to new fishes. On the other hand, CV-based approach studies are not required complexity analysis such as ripple activity and tracking analysis in which, small number of fishes and small tanks with special environment assist creating result. Tracking approach is applied to acknowledge movement of nutriment to understand more about the fish behavior. Fish behaviors can be obtained by combination between tracking analysis and ripple activity. Then, these fish behaviors can be a decision to start and stop fish feeding

machine by understanding of ripple activity after giving several nutriments. By explaining of fish behavior, tracking nutriment is important and it is required to analyze the complexity data in real environment.

By summarizing aforementioned problems, we proposed tuna nutriment tracking based on the classical minimum cost problem [50, 62, 81] where each detection calculates minimum distance among them and creates a trajectory to be tracked line. By collaborating with an active aquaculture fish farm, we develop tuna nutriment tracking using trajectory mapping. A video camera is placed above the boat with a highly disturbance of ocean wave and many dense nutriments. The camera captures between ocean surface and fish feeding machine. After that, videos transfer to a computer for further analysis the behavior of fish.

The aim of this research is tracking approach to acknowledge the behavior of tuna. For next, it can be useful to improve the production profit in fish farms by controlling the amount of nutriment in optimal rate.
To summarize, we make the following **contributions**:

- We propose tuna nutriment tracking based on trajectory mapping which can perform well as well as human annotator results.
- We propose a new novel small nutriment tracking method with collecting information of leading line into ripple.
- We show significantly improvement result of trajectory mapping in real environment datasets.

3.2 The Proposed Method - Trajectory Mapping

Our formulation is based on the classical minimum cost problem where each detection calculates minimum distance among them and creates a trajectory to be tracked line. In order to provide some background and formally introduce our approach, we start by providing flowchart and algorithm of tuna nutriment tracking. We then explain how the proposed method works to real environment. The proposed trajectory mapping contains a data normalization process, tuna nutriment detection and tuna nutriment tracking. The system flowchart of the proposed method is shown in Fig. 3.1, and the algorithm of the proposed trajectory mapping is represented in Algorithm 1 where D and $\mathbf{T}_f$ are input video and trajectory of time-ordered tuna nutriment, respectively.

Algorithm 1: Trajectory mapping algorithm

Input : D: a fish feeding video data.
Auxiliary methods : Data normalization, tuna nutriment detection, tuna nutriment tracking.
Output : Trajectory T_f

input a fish feeding video data D.
use image stabilization as data normalization $\hat{D} = \chi_\phi\{D\}$.
for $\varpi \leftarrow 1$ *to* n_f **do**
    calculate centroid of tuna nutriment detection $\sigma^c(\varpi)$ and ripple area detection $\sigma^r(\varpi)$.
    create an area of top-left $\alpha_\varpi = (\hat{x}^\alpha_\varpi, \hat{y}^\alpha_\varpi)$ and bottom right $\theta_\varpi = (\hat{x}^\theta_\varpi, \hat{y}^\theta_\varpi)$ of ripple detection.
    if $\varpi = 1$ **then**
        create initialization of upper limit trajectory $T^u_f(\varpi) = \{C_\varpi, \delta_\varpi, \alpha_\varpi\}$ and lower limit trajectory $T^l_f(\varpi) = \{C_{f_1}, \theta_\varpi\}$.
    end
    else if $\varpi = 2$ **then**
        update upper limit trajectory $T^u_f(\varpi) = \{C_{f_1}, C_{f_2}, \frac{\sum_{z=1}^{\varpi} \delta_z}{n_f}, \alpha_\varpi\}$.
    end
    else if $\varpi = 3$ **then**
        update upper limit trajectory $T^u_f(\varpi) = \{C_{f_1}, C_{f_2}, C_{f_3}, \frac{\sum_{z=1}^{\varpi} \delta_z}{\varpi}, \alpha_\varpi\}$.
        set trajectory of tuna nutriment prediction $T_f(\varpi) = T^u_f(\varpi)$.
    end
    else
        update trajectory of tuna nutriment prediction $T_f(\varpi) = \{C_{f_1}, C_{f_2}, \ldots, C_{f_\varpi}, \frac{\sum_{z=1}^{\varpi} \delta_z}{\varpi}, \alpha_\varpi\}$.
    end
end

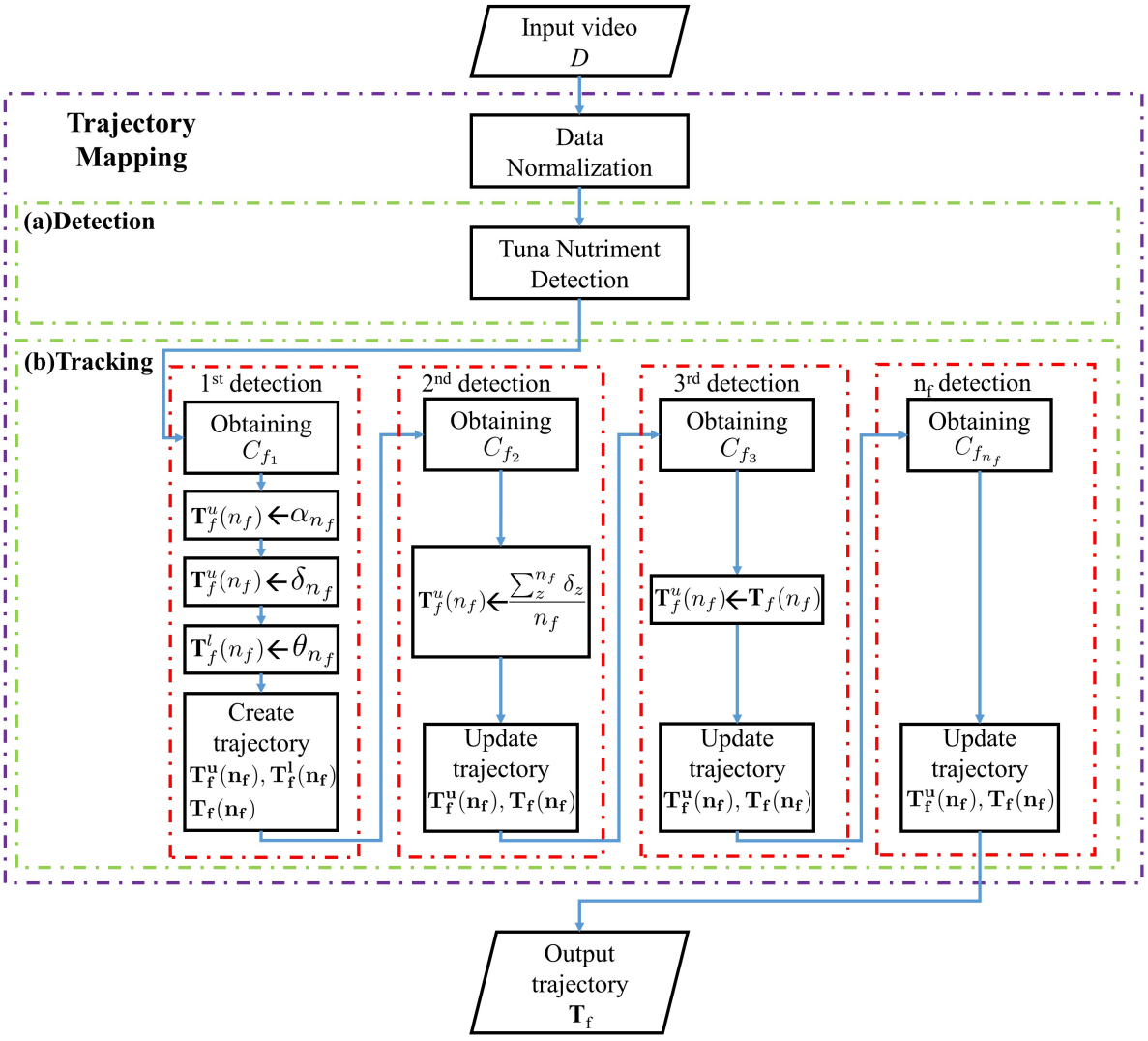


Fig. 3.1 Flowchart of the proposed trajectory mapping. The input video is received and applied image stabilization as data normalization. (a) Creating model for tuna nutriment detection using YOLOv3 [84] and obtaining bounding box for each tuna nutriment prediction in all frames of a video. (b) Tuna nutriment prediction as input tracking approach to be initialization for $n_f = 1$ detection. After that, we obtain upper and lower limit trajectory T_l^u, T_f^l and the maximum height of all tuna nutriment predictions δ_1. Next, we find the value of C_{f_2} using the shortest distance between C_{f_1} and all tuna nutriment predictions in n_f appearing in inside area between $T_f^u(1)$ and $T_f^u(1)$. Its process is repeatedly until n_f. Next, we obtain final trajectory as tracking result.

3.2.1 Data Normalization

For data normalization, image stabilization is applied to reduce a hand-held camera and ocean waves. Image stabilization is created by transformation from previous to current frame using

optical flow for all frames. [Ngh] accumulates rigid transformation χ to obtain linked between frame L. New rigid transformation χ_ϕ in frame ϕ can be written as:

$$\begin{aligned}\chi_\phi &= \chi_{\phi-1} + (\frac{1}{\gamma}\sum_{\tau=\phi-\gamma}^{\phi+\gamma} L_\tau)\check{}L_{\phi-1}, \\ \hat{D} &= \chi_\phi\{D\},\end{aligned} \tag{3.1}$$

where $\hat{D}$ is output video after applied image stabilization and γ is smoothing radius where the radius is number of frames used for smoothing and defined by 30.

3.2.2 Tuna Nutriment Detection

The idea of tuna nutriment detection is to produce boundary box in each nutriment associated in tracking method. In implementation of tuna nutriment detection, YOLOv3 [84] accumulates bounding box of tuna nutriment prediction $B = (\hat{x}, \hat{y}, \hat{w}, \hat{h})$ by training model with bounding box $P = (p_x, p_y, p_w, p_h)$ of ground truth data where p_x, p_y, p_w, and p_h are centroid x, centroid y, width, and height of bounding box in ground truth data, respectively. ς_x and ς_y represent the absolute location of the top-left corner of the current grid cell. w and h are the absolute width and height to the whole image. Bounding box of tuna nutriment prediction B can defined as:

$$\begin{aligned}\hat{x} &= \delta(p_x) + \varsigma_x \\ \hat{y} &= \delta(p_y) + \varsigma_y \\ \hat{w} &= e^{p_w} * w \\ \hat{h} &= e^{p_h} * h\end{aligned} \tag{3.2}$$

where δ is model followed by [84].

3.2.3 Tuna Nutriment Tracking

In order to represent tracking of tuna nutriment, introducing how to collect set of tuna nutriment prediction corresponding to time-ordered path in the graph is important. We are given $\sigma^c(n) = \{C_1, C_2, C_3, \ldots, C_n\}$ as input centroid of tuna nutriment predictions where n is the total number of nutriment for all frames of video $\hat{D}$. Each tuna nutriment prediction is represented by $C_n = \{\hat{x}_n^c, \hat{y}_n^c\}$. Definition of a trajectory is denoted as centroid of time-ordered tuna nutriment predictions $T_f(n_f) = \{C_{f_1}, C_{f_2}, C_{f_3}, \ldots, C_{f_{n_f}}\}$ where n_f is the number of detections formed

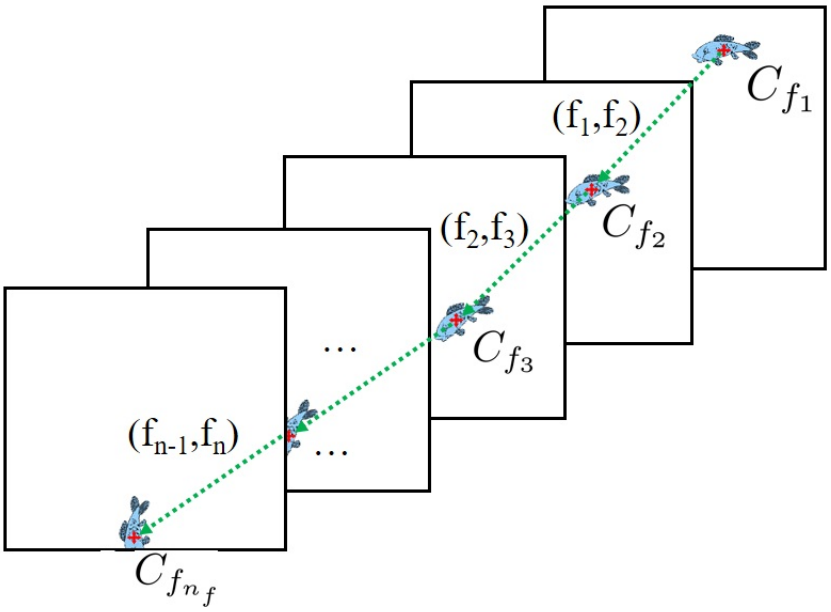


Fig. 3.2 Visualization of trajectory of tuna nutriment predictions $T_f = \{C_{f_1}, C_{f_2}, C_{f_3}, \ldots, C_{f_{n_f}}\}$ in which every tuna nutriment predictions is connected by node $\{(f_1, f_2), (f_2, f_3), \ldots, (f_{n-1_f}, f_{n_f})\}$.

by trajectory f. So that, $\rho = \{n_1, n_2, n_3, \ldots, n_{n_f}\}$ can be denoted as the total of number of nutriments appearing in every time-ordered trajectory $T_f(n_f)$.

Problem Statement

The problem can be represented with an undirected graph $G = (V, E)$, where $V := \{1, ..., n\}, E \subset V^2$, and each node $f \in V$ denotes a unique detection $C_f \in \sigma^c$. The task of dividing the set of tuna nutriment predictions into trajectories can be observed as grouping nodes in graph. Fig. 3.2 shows that each trajectory $T_f(n_f) = \{C_{f_1}, C_{f_2}, C_{f_3}, \ldots, C_{f_{n_f}}\}$ in the scene can be mapped into a group of nodes $\{(f_1, f_2), (f_2, f_3), \ldots, (f_{n-1_f}, f_{n_f})\}$. To produce each $C_{f_{[1,n_f]}}$, trajectory mapping is applied in next section.

In two-dimensional trajectory, the component of trajectory is divided by horizontal and vertical direction. In vertical direction, acceleration is constant and has quadratic function. Trajectory mapping applies the idea of acceleration and chooses quadratic function as basis.

To produce quadratic function $y^c = a_3^c x^2 + a_2^c x + a_1^c$ as a result of trajectory T_f, we apply polynomial fitting [Weisstein] defined by calculation of $\hat{x}_{n_f}^c$ to form Vandermonde matrix V with 3 columns as results of a^c.

$$\begin{bmatrix} 1 & \hat{x}_1^c & (\hat{x}_1^c)^2 \\ 1 & \hat{x}_2^c & (\hat{x}_2^c)^2 \\ 1 & \vdots & \vdots \\ 1 & \hat{x}_{n_f}^c & (\hat{x}_{n_f}^c)^2 \end{bmatrix} \begin{bmatrix} a_1^c \\ a_2^c \\ a_3^c \end{bmatrix} = \begin{bmatrix} \hat{y}_1^c \\ \hat{y}_2^c \\ \vdots \\ \hat{y}_{n_f}^c \end{bmatrix} \tag{3.3}$$

(3.3) can be inverted directly. To yield the solution vector a^c, it can be defined as:

$$a^c = (X^T X)^{-1} X^T Y, \tag{3.4}$$

Tuna Nutriment Predictions $C_{f_{n_f}}$ where $n_f = 1$ as Initialization Point Detection

Tuna nutriment predictions C_{f_1} are obtained from every tuna nutriment prediction C_n in around cutting area of w. To define cutting area, we use centroid $\hat{x}_{f_1}$ as component of C_{f_1} by thresholding in w which is defined as:

$$w * \gamma \leq \hat{x}_{f_1} \leq w, \tag{3.5}$$

where γ is an input parameter and empirically defined as 0.9.

Direction of nutriment is calculated by leading nutriment to ripple area around sea levels. We are given a pair set of ripple area detection $\sigma^r(n_f) = \{(R_{11}, R_{12}), (R_{21}, R_{22}), (R_{31}, R_{32}), \ldots, (R_{n_f 1}, R_{n_f 2})\}$ as time-ordered ripple predictions in number of detections n_f. Each ripple prediction is represented by $R_{n_f 1,2} = \{\hat{x}_{n_f}^r, \hat{y}_{n_f}^r, \hat{w}_{n_f}^r, \hat{h}_{n_f}^r\}$. We divide component of ripple prediction to be an area of top-left $\alpha_{n_f} = (\hat{x}_{n_f}^\alpha, \hat{y}_{n_f}^\alpha)$ and bottom right $\theta_{n_f} = (\hat{x}_{n_f}^\theta, \hat{y}_{n_f}^\theta)$ of ripple detection by following:

$$\begin{aligned} \hat{x}_{n_f}^\alpha &= \hat{x}_{n_f}^r - \frac{\hat{w}_{n_f}^r}{2}, \\ \hat{y}_{n_f}^\alpha &= \hat{y}_{n_f}^r - \frac{\hat{h}_{n_f}^r}{2}, \\ \hat{x}_{n_f}^\theta &= \hat{x}_{n_f}^r + \frac{\hat{w}_{n_f}^r}{2}, \\ \hat{y}_{n_f}^\theta &= \hat{y}_{n_f}^r + \frac{\hat{h}_{n_f}^r}{2}, \end{aligned} \tag{3.6}$$

Trajectory Mapping

To obtain more feature, we need to know possibly coverage area for possibly nutriment appearing in next frame by creating upper and lower limit trajectory T_f^u and T_f^l, respectively. Upper and lower limit trajectory $T_f^u(n_f)$ and $T_f^l(n_f)$ formed by trajectory f are initialized by following:

$$\begin{aligned} T_f^u(n_f) &= \{C_{f_1}, \delta_{n_f}, \alpha_{n_f}\}, \\ T_f^l(n_f) &= \{C_{f_1}, \theta_{n_f}\}, \end{aligned} \tag{3.7}$$

where δ is the maximum height of all nutriment detections in n_f. (3.7) can be simplify by substituting $n_f = 1$ to be:

$$\begin{aligned} T_f^u(1) &= \{C_{f_1}, \delta_1, \alpha_1\}, \\ T_f^l(1) &= \{C_{f_1}, \theta_1\}, \end{aligned} \tag{3.8}$$

where $\delta_1 = (\frac{\hat{x}_1^{\alpha} + w}{2}, \min\limits_{1 \le z \le n_1} \hat{y}_z^c)$.

Tuna Nutriment Predictions $C_{f_{n_f}}$ where $n_f = 2$

To be a candidate of C_{f_2}, we use all tuna nutriment predictions n appearing in the inside of area between $y^u = a_3^u x^2 + a_2^u x + a_1^u$ and $y^l = a_3^l x^2 + a_2^l x + a_1^l$. Vector a^u and a^l are produced by calculating $T_f^u(n_f - 1)$ and $T_f^l(n_f - 1)$ with Vandermonde matrix shown in (3.3) and (3.4), respectively. Given $\sigma^{\kappa}(\mu) = \{\kappa_{1_{n_f}}, \kappa_{2_{n_f}}, \kappa_{3_{n_f}}, \ldots, \kappa_{\mu_{n_f}}\}$ is a set of candidate $C_{f_{n_f}}$. $C_{f_{n_f}}$ is defined by the nutriment predictions which have shortest distance denoted by:

$$\begin{aligned} C_{f_{n_f}} \quad = \quad & \arg\min_{\mu} (Z(f_{n_f-1}) - \sigma^{\kappa}(\mu))^T \\ & (Z(f_{n_f-1}) - \sigma^{\kappa}(\mu)) \end{aligned} \tag{3.9}$$

where $Z(f_{n_f-1}) = \{C^1_{f_{n_f-1}}, C^2_{f_{n_f-1}}, C^3_{f_{n_f-1}}, \ldots, C^{\mu}_{f_{n_f-1}}\}$. (3.9) can be simplify to be;

$$C_{f_2} = \arg\min_{\mu} (Z(1) - \sigma^{\kappa}(\mu))^T (Z(1) - \sigma^{\kappa}(\mu)), \tag{3.10}$$

Updating upper trajectory $T_f^u(n_f)$ can be defined as:

$$T_f^u(n_f) = \{C_{f_1}, C_{f_2}, \frac{\sum_{z=1}^{n_f} \delta_z}{n_f}, \alpha_{n_f}\}, \tag{3.11}$$

Tuna Nutriment Predictions $C_{f_{n_f}}$ where $n_f = 3$

Minimum requirement for trajectory of quadratic functions must have at least 3 tuna nutriment predictions collected. To produce C_{f_3}, (3.9) is applied using $n_f = 3$ as parameter. Then, updating upper limit trajectory $T_f^u(n_f)$ is denoted as follows:

$$T_f(n_f) = T_f^u(n_f), \tag{3.12}$$

Tuna Nutriment Predictions $C_{f_{n_f}}$ where $n_f \geq 4$

To precise accuracy of trajectory T_f, we refine its trajectory by collecting more tuna nutriment prediction $C_{f_{n_f}}$. Tuna nutriment prediction $C_{f_{n_f}}$ is calculated using the nearest nutriment detection in area of $y^u = a_3^u x^2 + a_2^u x + a_1^u$ with tolerance degree from quadratic function between ± 30 degree.

To handle losing tuna nutriment prediction, we used previously tuna nutriment prediction by calculating the speed of nutriment in next frame.

$$\begin{aligned} \hat{x}_{f_{n_f}}^c &= 3\hat{x}_{f_{n_f-1}}^c - 3\hat{x}_{f_{n_f-2}}^c + \hat{x}_{f_{n_f-3}}^c, \\ \hat{y}_{f_{n_f}}^c &= a_3^c(\hat{x}_{f_{n_f}}^c)^2 + a_2^c \hat{x}_{f_{n_f}}^c + a_1^c, \end{aligned} \tag{3.13}$$

where a_1^c, a_2^c, and a_3^c are coefficients of quadratic function formed by trajectory T_f

3.3 Experiment

In this section, we first explain the details of our datasets. We then describe evaluation approach to calculate error rate distance and show quantitative evaluation with various of n_f to discover an optimal value.

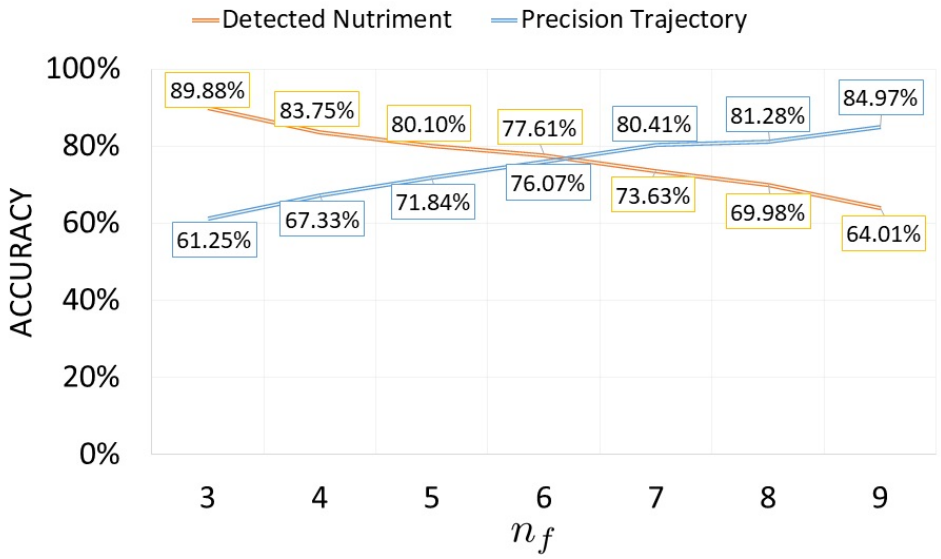


Fig. 3.3 Accuracy detected nutriment and precision trajectory in various $n_f \in [3,9]$. The optimum value of detected nutriment and precision trajectory is $n_f = 6$.

3.3.1 Evaluation Approach

Evaluation approach is defined by measuring minimum euclidean distance based on number of nutriment collected with ground truth T_g. Best trajectory T^* with minimum error rate distance is defined as:

$$T^* = \arg\min_{n_f} (T_g - T(n_f))^T (T_g - T(n_f)), \tag{3.14}$$

where $n_f \in [3,9]$.

3.3.2 Quantitative Evaluation with various of $n_f = [3,9]$

Quantitative evaluation is computed by performance of detected nutriment and precision trajectory showed in Fig. 3.3. Number of detected nutriment is defined as percentage of detected nutriment divided by ground truth of nutriments appearing in frame. Meanwhile, precision trajectory is computed by total number of nutriments having trajectory leading to ripple area divided by detected nutriment. Fig. 3.4 and Table 3.1 show the confidence interval and statistical analysis of error rate distance in various of n_f. The results show that the optimal value of n_f is 6 in which this parameter produces smallest error rate distance.

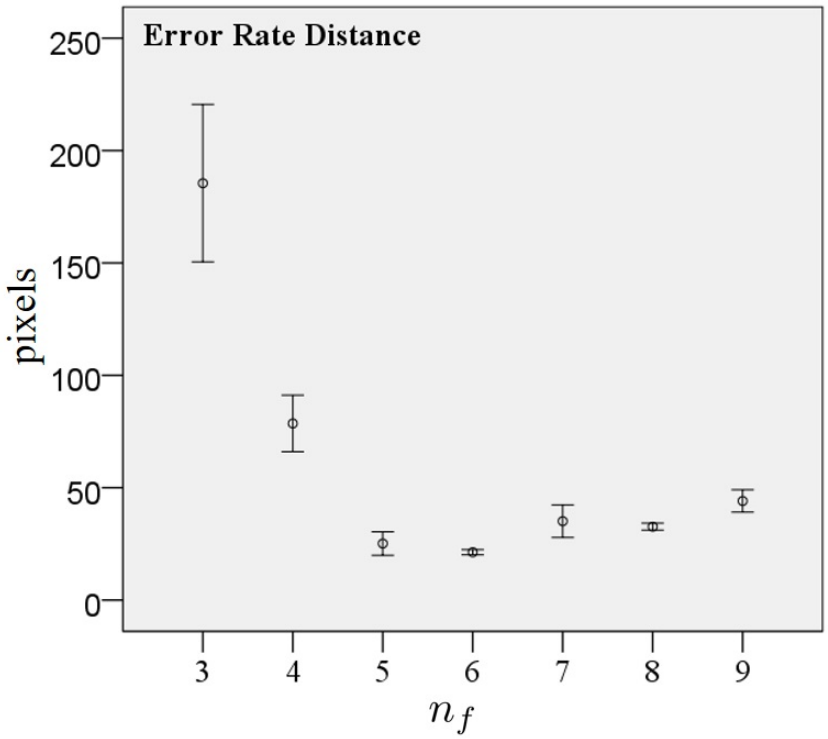


Fig. 3.4 95% Confidence interval of error rate distance in various of $n_f \in [3,9]$. The lowest error rate distance of proposed method is $n_f = 6$.

3.4 Result

In this section, we compare proposed method and state-of-the-art benchmark methods on our datasets. After that, we show the figures to explain the advantage of the proposed method and computational time between proposed method and state-of-the-art benchmark methods.

3.4.1 Evaluation Result

Precision of mAP in object detection is computed by performance YOLOv3 [84] to train our datasets with 10k iterations with 416×416 pixels for image resizing from 1920×1080 pixels. Fig. 3.5 displays training result of our datasets using YOLOv3 [84] and reaches 67% of maximum mAP with 10k iterations. We also tested proposed methods and state-of-the-art benchmark results. There are many state-of-the-art methods using multiple object tracking (MOT) [105, 64, 101, 117, 113, 12, 78]. These methods perform well using six publicly available datasets on pedestrian detection, MOT and person search provided by [71, 57, 30]. In evaluations, we choose JDE [105] to represent MOT as benchmark method because JDE is very fast and accurate based on re-implementation of faster object detection compared with

Table 3.1 Statistical analysis of various $n_f \in [3,9]$ for quantitative evaluation using one-samples T-Test.

nf	n	Mean (pixels)	Std. Dev. (pixels)	Std. Error (pixels)	95% Confidence Interval of the Diff.	
					Lower (pixels)	Upper (pixel)
3	30	185.47	93.81	17.13	150.44	220.5
4	30	78.58	33.64	6.14	66.02	91.14
5	30	25.17	13.97	2.55	**19.96**	30.39
6	30	**21.32**	**3.08**	**0.56**	20.18	**22.48**
7	30	35.12	19.34	3.53	27.9	42.35
8	30	32.67	4.19	0.76	31.11	34.24
9	30	44.1	13.32	2.43	39.12	49.07

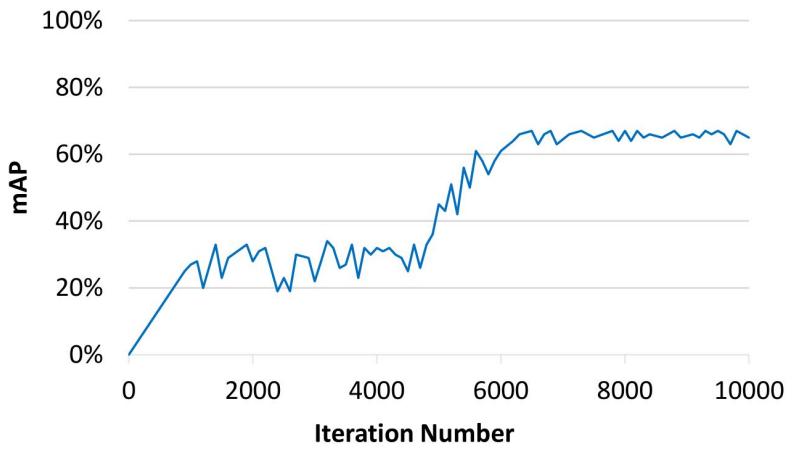


Fig. 3.5 mAP result during training model using YOLOv3 [84]. After 7k iterations, curve of mAP of model has more stable.

[64, 101, 117, 113, 12, 78]. We also use SORT [16] as benchmark methods and add our detection model to completely understand performance of tracking method.

In Fig. 3.6a, the proposed method is demonstrated to be able to track small nutriment while JDE and SORT with original YOLOv3 and our detection model perform poor (Fig. 3.6b, 3.6c, 3.6d) without tracking results of nutriments even SORT is able to detect some nutriments. By our experiment, the benchmark methods fail to run our datasets because the size of nutriment is too small (maximum size is 13×36 pixels) and the speed of nutriment is fast (average nutriment movement from start to end node is 23.8 frames).

Table 3.2 Hardware and software environment for running proposed method and benchmark methods for comparison.

Spesification		
Hardware	CPU	Intel Core i7-9700 CPU @3.00GHz (8 CPUs)
	RAM	16 GB
	GPU	NVIDIA GeForce GTX 745
Software	OS	Windows 10 Pro 64-bit
	IDE	Microsoft Visual Studio Professional 2017 v.15.9.25
	Language	Python 3.6 64bit

Table 3.3 Comparison computational time proposed method and benchmark methods.

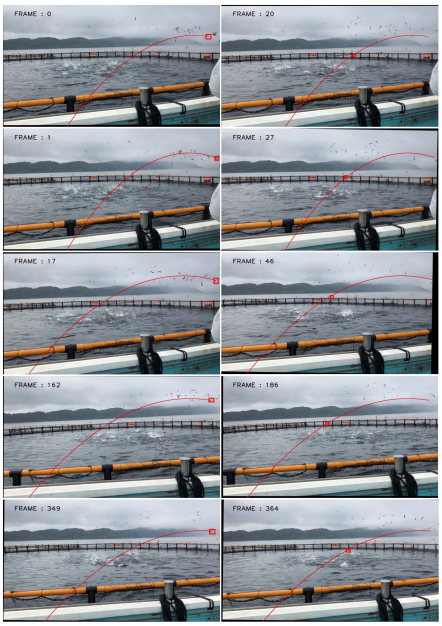
Methods	N	Mean (fps)	Std. Dev. (fps)	Std. Err. (fps)	95% Confidence Interval of the Diff.	
					Lower (fps)	Upper (fps)
Ours	419	1.93	0.61	0.03	1.87	1.99
JDE	419	1.87	0.07	0.00	1.86	1.87
YOLOv3 + Sort	419	0.45	-	-	-	-
Our Detection + Sort	419	0.47	-	-	-	-

3.4.2 Implementation Details

For analysis of the computational complexity and execution time of the proposed methodology, a computational time analysis is conducted using a video with 419 frames. Table 3.2 shows the specification of hardware and software for comparison. Table 3.3 compares the computation time (in fps) for proposed method, namely trajectory mapping and benchmark approaches: JDE and SORT with original YOLOv3 and our detection model. For average and standard deviation of computational time, we reach 1.93 and 0.61 fps, while JDE spends 1.87 and 0.07 fps, respectively. SORT only provides average computational time without information of computational time for individual frame. Computational time for both detection model of YOLOv3 and our detection model with SORT performs worst and these benchmark approaches reach 0.45 fps and 0.47, respectively. By analyzing computational complexity, proposed method runs faster than JDE with the different speed is 0.6 fps.

3.5 Conclusion and Discussion

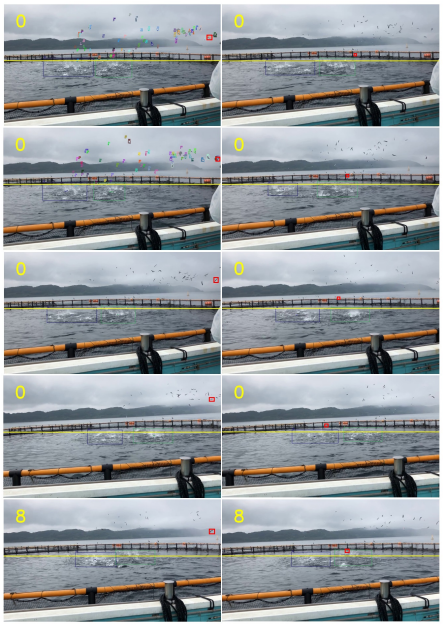
Tracking approach is the one of features to analyze fish behavior to create a decision to optimize the amount of nutriment. Recent studies have shown that it is possible to track movement objects in entire of frames on video. However, there is no agreement to track multiple small nutriments in the video which has interference of hand-held camera and ocean waves. In this paper, tuna nutriment tracking using trajectory mapping in application to aquaculture fish tank has been presented and demonstrated to be promising for interference video containing multiple small nutriment datasets. We have demonstrated tuna nutriment tracking using trajectory mapping and the method consistently performs well on the interference video with good precision trajectory result. We expect our approach to open the door for future work and to go beyond for feature extraction of ripple activity and focus on integrating tracking approach and ripple activity to be a decision to control fish feeding machine.



(a) Trajectory Mapping

(b) JDE

(c) YOLOv3+SORT



(d) Our Detection Model + SORT

Fig. 3.6 Observation of proposed method namely trajectory mapping shown in (a) and benchmark method results shown in (b), (c), and (d). Left and right images for each method represent first node of nutriment and nutriment tracked after several frames, respectively. In trajectory mapping method, red curve is defined as trajectory result of proposed method. Red box in each image represents ground truth of nutriment. We can see that both red curve and red box in trajectory mapping show precisely tracked result and it proves that trajectory mapping creates trajectory very well while benchmark methods perform poor without tracking results of nutriments even SORT is able to detect some nutriments.

Chapter 4

Regression

4.1 Introduction

Raising fish utilization to complete the demand of human and animal resourcing in worldwide drives researchers exploring and improving technology in aquaculture industry [98, 5, 49]. Based on global aquaculture production statistic database [FAO], the proportion of aquatic animals farmed is 55.1 million tonnes in 2009 and significantly increasing to 82.1 million tonnes in 2018. This trend indicates that aquaculture industry has to create new technique to enlarge economic scale with reducing production cost and increasing production efficiency.

In this chapter, we present an automatic controlling fish feeding machine using combination of prediction nutriment position in next frame. Our method is constructed by object detection of nutriment. Regression model is used to predict each detection of nutriment in next frame. Regression can be useful to improve the production profit in fish farms by controlling the amount of nutriment in optimal rate and to optimize the use of fish feeding machine. The datasets are extracted from a video camera which is set down above vessel with a highly distraction of ocean wave and innumerable small and dense nutriments. After that, the camera captures above sea surface and the videos transfer to a computer for further analysis.

4.2 Proposed Method

Our formulation is based on tracking algorithm using regression where we estimate each nutriment detection in next frame. In order to provide some background and formally introduce our approach, we start by providing diagram and algorithm of regression approaches. We then explain how the proposed method works to real environment. The proposed method contains two parts: detection and regression which are shown in Figure 4.1 where the detection result

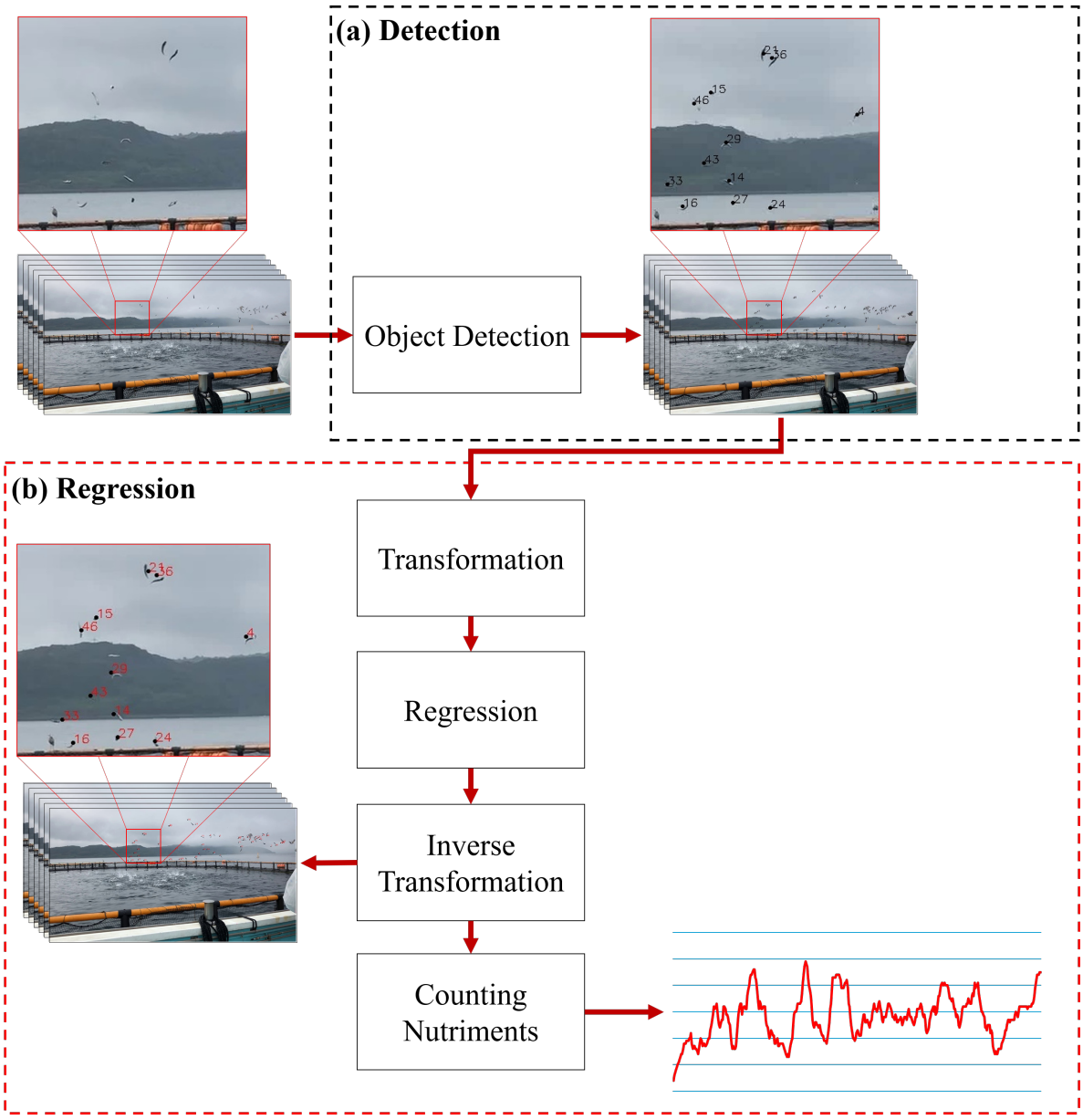


Fig. 4.1 Diagram of the proposed method. (a) The input video is received and applied object detection using YOLOv4 [17]. (b) Creating model for predicted tuna nutriment in next frame using regression. Before and after regression process, we calculate transformation and inverse transformation to create robust result.

Algorithm 2: Regression and textural feature of ripple behavior algorithm

Input : D: a fish feeding video data.
Auxiliary methods : Tuna nutriment detection, transformation, inverse transformation, regression, counting nutriments.
Output : A graph of regression

input a fish feeding video data D.
for $\vartheta \leftarrow 1$ *to* f **do**
  calculate bounding box of tuna nutriment detection $\varphi^{t}(\vartheta)$.
  create an area of top-left points $tl_{\vartheta 1,2} = (\hat{x}^{tl}_{\vartheta}, \hat{y}^{tl}_{\vartheta})$, top-right points $tr_{\vartheta 1,2} = (\hat{x}^{br}_{\vartheta}, \hat{y}^{tl}_{\vartheta})$, bottom-left points $bl_{\vartheta 1,2} = (\hat{x}^{tl}_{\vartheta}, \hat{y}^{br}_{\vartheta})$ and bottom-right points $br_{\vartheta 1,2} = (\hat{x}^{br}_{\vartheta}, \hat{y}^{br}_{\vartheta})$ of ripple detection.
  compute normalized nutriments $\varphi^{\kappa}(\vartheta)$ from geometry transformation $\varphi^{\xi}(\vartheta)$ and rotated nutriments $\varphi^{\Psi}(\vartheta)$.
  predict position of normalized nutriments $\varphi^{\kappa}(\vartheta+1)$ in next frame using regression model μ^{x} and μ^{y}.
  transform the result of regression $\varphi^{\kappa}(\vartheta+1)$ into original pixel position in real images $\varphi^{t}(\vartheta+1)$ using rotated nutriments $\varphi^{\Psi}(\vartheta+1)$ and geometry transformation $\varphi^{\xi}(\vartheta+1)$.
  compute passed line $y^{l}(\vartheta) = \alpha_{\vartheta} x^{l}_{\vartheta} + \beta_{\vartheta}$ to count the nutriment.
end

is represented as black color value and predicted result of regression is shown by red color value. Regression contains both transformation and inverse transformation, regression approach and counting nutriments. We also present the algorithm of the proposed method shown in Algorithm 2.

4.2.1 Object Detection

The idea of object detection is to find bounding box in each nutriment and ripple area associated in regression and textural feature. In implementation of object detection, YOLOv4 [17] has different part of object detector: backbone, neck, and head. Backbones uses CSPDarknet53 as feature-extractor model and are more suitable for classification than for detection. Necks utilize spatial pyramid pooling (SPP) which is pooling layer removing fixed size constraint of network and path aggregation network (PAN) which aims to improve the information flow of segmentation network. The latter of necks architecture has different forms to the original PAN and it is modified version with replacing the addition layer with a concat. On the other hand, the head of YOLOv4 [17] has similar form with YOLOv3 [84] which computes training model with bounding box of object detection $B = (\hat{x}^{b}, \hat{y}^{b}, \hat{w}^{b}, \hat{h}^{b})$ and bounding box $G = (x^{g}, y^{g}, w^{g}, h^{g})$ of ground truth data where x^{g}, y^{g}, w^{g}, and h^{g} are center x, center y, width, and height of bounding

box in ground truth data, respectively. λ_x and λ_y represent the absolute top-left corner location of the current grid cell. w and h are width and height referenced to size of image. Bounding box prediction B can defined as:

$$\begin{aligned}\hat{x}^b &= \zeta(x^g) + \lambda_x, \\ \hat{y}^b &= \zeta(y^g) + \lambda_y, \\ \hat{w}^b &= e^{w^g} * w, \\ \hat{h}^b &= e^{h^g} * h,\end{aligned} \tag{4.1}$$

where ζ is model reffed to [17].

4.2.2 Regression Approach

Given $\varphi^t(f) = \{T_{f1}, T_{f2}, \cdots, T_{fa}\}$ is tuna nutriment detection a in frame f. For each detection, it is represented by $T_{fa} = (\hat{x}^t_{fa}, \hat{y}^t_{fa}, \hat{w}^t_{fa}, \hat{h}^t_{fa})$. Each bounding box of tuna nutriment detection occurs center of bounding box $\hat{x}^t_{fa}$, center of bounding box $\hat{y}^t_{fa}$, width $\hat{w}^t_{fa}$, and height $\hat{h}^t_{fa}$. For giving robust solution, center of tuna nutriment $\hat{x}^t_{fa}$ and $\hat{y}^t_{fa}$ is transformed to different domain to reduce rotation and unknown position of camera effect. Regression is used to predict tuna nutriment $\hat{x}^t_{fa}$ and $\hat{y}^t_{fa}$ in next frame $f+1$. The results of regression apply inverse transformation for moving back to original center points of predicted nutriments $\hat{x}_{(f+1)a}$ and $\hat{y}_{(f+1)a}$.

Transformation

We are given a pair of ripple area detection $\varphi^r(f) = \{R_{f1}, R_{f2}\}$ in frame f. Each ripple detection is represented by $R_{f1,2} = (\hat{x}^r_f, \hat{y}^r_f, \hat{w}^r_f, \hat{h}^r_f)$. We compute component of ripple detection to be top-left points $tl_{f1,2} = (\hat{x}^{tl}_f, \hat{y}^{tl}_f)$, top-right points $tr_{f1,2} = (\hat{x}^{br}_f, \hat{y}^{tl}_f)$, bottom-left points $bl_{f1,2} = (\hat{x}^{tl}_f, \hat{y}^{br}_f)$ and bottom-right points $br_{f1,2} = (\hat{x}^{br}_f, \hat{y}^{br}_f)$ of ripple detection by following:

$$\begin{aligned}\hat{x}^{tl}_f &= \frac{2\hat{x}^r_f - \hat{w}^r_f}{2}, \\ \hat{y}^{tl}_f &= \frac{2\hat{y}^r_f - \hat{h}^r_f}{2}, \\ \hat{x}^{br}_f &= \frac{2\hat{x}^r_f + \hat{w}^r_f}{2}, \\ \hat{y}^{br}_f &= \frac{2\hat{y}^r_f + \hat{h}^r_f}{2}.\end{aligned} \tag{4.2}$$

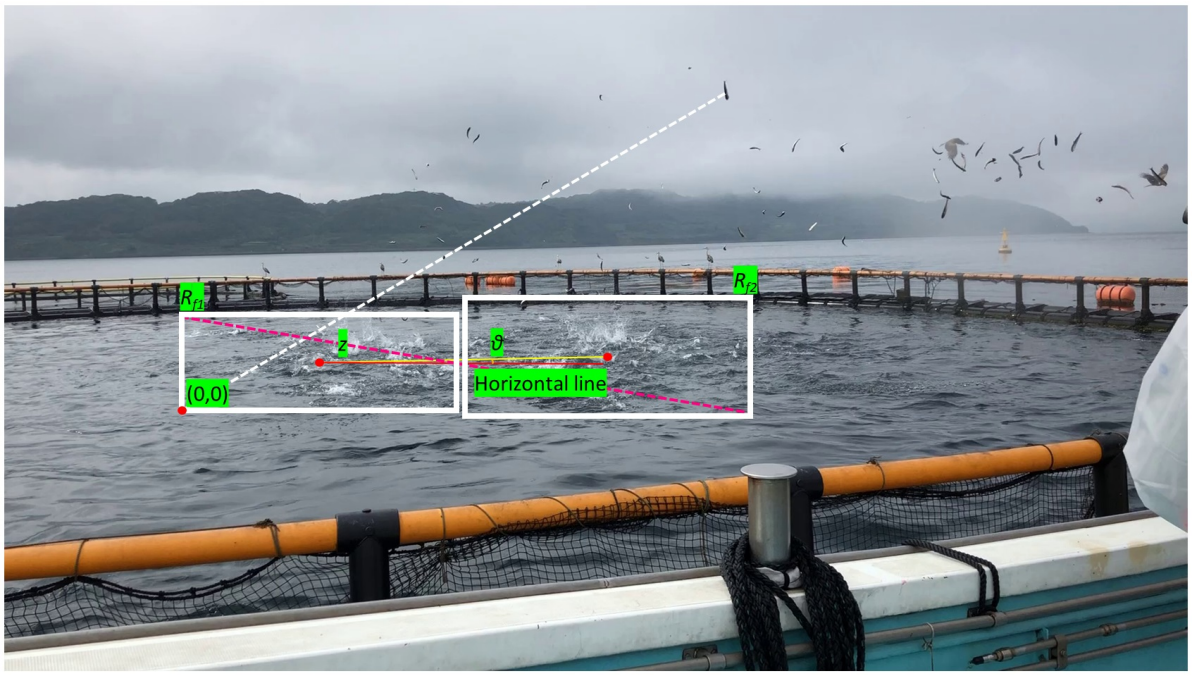


Fig. 4.2 Visualization of transformation and inverse transformation approaches. White bounding box is represented as ripple detection. The center of coordinate position $(0,0)$ after transformed is shown in bottom-left point lb_{f1} of ripple R_{f1}. The distance from the center of coordinate position $(0,0)$ to nutriment is shown by white line. Angle θ is obtained from the difference between two center points of ripple and horizontal line. Pink line is computed by top-left points tl_{f1} of ripple R_{f1} and bottom-right points br_{f2} of ripple R_{f2}.

Regression

After calculating the corner of ripple detection, normalizing coordinate of nutriment is important aspect to give robust solution. The idea behind the coordinate transformation of nutriments is that the video camera moves around and is also that there is no distance information between camera and ripple location. The goal of normalizing coordinate is to transform the function $\varphi^t(f)$ of nutriments into $\varphi^\kappa(f) = \{\kappa_{f1}, \kappa_{f2}, \cdots, \kappa_{fa}\}$ as normalized of nutriments. We assume that the ripple location can be defined as the parameter to acknowledge moving of camera and changing the distance between camera and ripple area. Bottom-left points lb_{f1} of ripple R_{f1} is the new center point of normalizing coordinate with moving the position of $(0,0)$ pixel in image shown in Figure 4.2. To compute angle θ_f of both ripples, we use center of R_{f1} and R_{f2} which is defined as follow:

$$\theta_f = \arctan_2((\hat{y}^r_{f2} - \hat{y}^r_{f1}), (\hat{x}^r_{f2} - \hat{x}^r_{f1})), \tag{4.3}$$

Angle θ_f is used to rotate center of tuna nutriment detection $(\hat{x}^t_{fa}, \hat{y}^t_{fa})$ by bottom-left points $(\hat{x}^{tl}_{f1}, \hat{y}^{br}_{f1})$ of ripple R_{f1} which is presented by:

$$\begin{aligned} \hat{x}^\psi_{fa} =& (\hat{x}^t_{fa} - \hat{x}^{tl}_f)\cos(\theta_f) - (\hat{y}^t_{fa} - \hat{y}^{br}_f)\sin(\theta_f) + \hat{x}^{tl}_f, \\ \hat{y}^\psi_{fa} =& (\hat{x}^t_{fa} - \hat{x}^{tl}_f)\sin(\theta_f) + (\hat{y}^t_{fa} - \hat{y}^{br}_f)\cos(\theta_f) + \hat{y}^{br}_f, \end{aligned} \tag{4.4}$$

where $\psi_{fa} = (\hat{x}^\psi_{fa}, \hat{y}^\psi_{fa})$ is rotating result from a rotated function of nutriments $\varphi^\psi(f) = \{\psi_{f1}, \psi_{f2}, \cdots, \psi_{fa}\}$. After normalizing using rotation, translation is applied to change the center point from the position of $(0,0)$ pixel in image to bottom-left points bl_{f1} of ripple R_{f1} shown in Figure 4.2 which is defined as follow:

$$\begin{aligned} \hat{x}^\xi_{fa} =& \hat{x}^\psi_{fa} - \hat{x}^{tl}_{f1}, \\ \hat{y}^\xi_{fa} =& \hat{y}^{br}_{f1} - \hat{y}^\psi_{fa}, \end{aligned} \tag{4.5}$$

where $\xi_{fa} = (\hat{x}^\xi_{fa}, \hat{y}^\xi_{fa})$ is geometry transformation result after rotation from a geometry transformation function of nutriments $\varphi^\xi(f) = \{\xi_{f1}, \xi_{f2}, \cdots, \xi_{fa}\}$. Distance z is computed by top-left points tl_{f1} of ripple R_{f1} and bottom-right points br_{f2} of ripple R_{f2} shown by:

$$z = \sqrt{(\hat{x}^{tl}_{f1} - \hat{x}^{br}_{f2})^2 + (\hat{y}^{tl}_{f1} - \hat{y}^{br}_{f2})^2}. \tag{4.6}$$

Normalized of nutriment $\kappa_{fa} = (\hat{x}^{\kappa}_{fa}, \hat{y}^{\kappa}_{fa})$ is defined as fraction of geometry transformation ξ_{fa} with distance z which is explained by:

$$\begin{aligned} \hat{x}^{\kappa}_{fa} &= \frac{\hat{x}^{\xi}_{fa}}{z}, \\ \hat{y}^{\kappa}_{fa} &= \frac{\hat{y}^{\xi}_{fa}}{z}. \end{aligned} \tag{4.7}$$

Inverse Transformation

Inverse transformation is used to move back to original center points of nutriments $\hat{x}_{fa}$ and $\hat{y}_{fa}$. This process is reverse transformation process from a function of normalized nutriment $\varphi^{\kappa}(f)$ to $\varphi^{t}(f)$. The inverse transformation is started from normalized of nutriments $\varphi^{\kappa}(f)$ to geometry transformation $\varphi^{\xi}(f)$ presented by:

$$\begin{aligned} \hat{x}^{\xi}_{fa} &= \hat{x}^{\kappa}_{fa} * z, \\ \hat{y}^{\xi}_{fa} &= \hat{y}^{\kappa}_{fa} * z, \end{aligned} \tag{4.8}$$

After getting geometry transformation $\varphi^{\xi}(f)$, translation is applied to convert from a function of geometry transformation $\varphi^{\xi}(f)$ to rotated function $\varphi^{\psi}(f)$ which is represented by:

$$\begin{aligned} \hat{x}^{\psi}_{fa} &= \hat{x}^{\xi}_{fa} + \hat{x}^{tl}_{f1}, \\ \hat{y}^{\psi}_{fa} &= \hat{y}^{br}_{f1} - \hat{y}^{\xi}_{fa}, \end{aligned} \tag{4.9}$$

Our goal is transforming a function of rotated nutriment $\varphi^{\psi}(f)$ to original pixel position in real images $\varphi^{t}(f)$ using (4.4) with negative of angle θ_f presented by:

$$\theta_f = -\arctan_2((\hat{y}^{r}_{f2} - \hat{y}^{r}_{f1}).(\hat{x}^{r}_{f2} - \hat{x}^{r}_{f1})), \tag{4.10}$$

Regression

Small and dense nutriment can easily fail to be detected for several frames because current position and trajectory of each individual nutriment have different angle and speed. To propose robust solution for undetected nutriments for several frames, regression can be superior algorithm rather than tracking algorithm because performance of tracking depends on nutriment

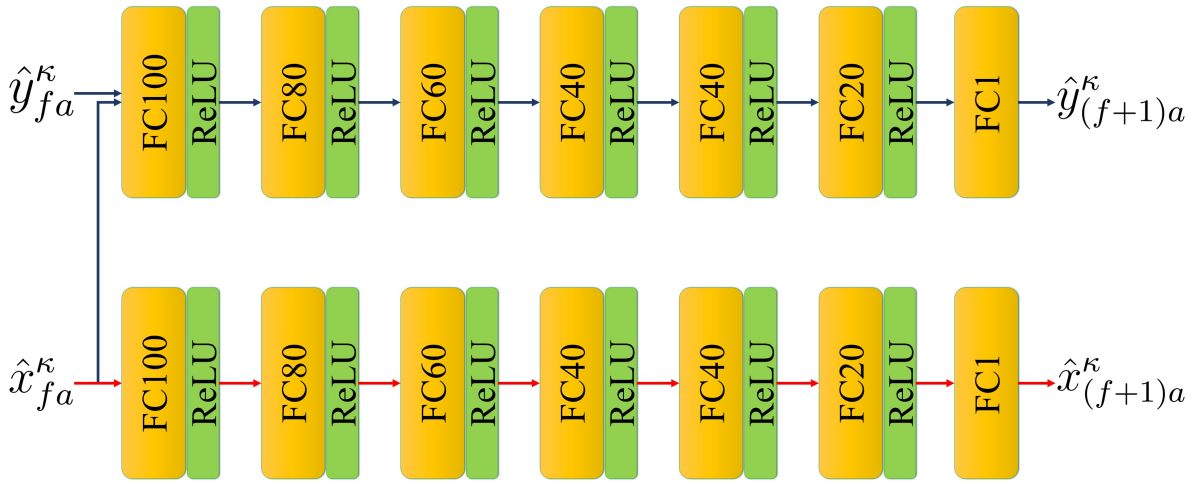


Fig. 4.3 Structure network of regression for predicting all nutriments detection in next frame.

detection results in sequential frames. To improve regression result, the complexity of network by using multiple neuron and hidden layer is needed to precisely accurate prediction the moving nutriment with lower error direction. Model of regression μ^x and μ^y predicts the nutriments from current frame to next position of nutriment $(\hat{x}^{\kappa}_{(f+1)a}, \hat{y}^{\kappa}_{(f+1)a})$ in next frame which is defined as:

$$\begin{aligned} \hat{x}^{\kappa}_{(f+1)a} &= \mu^x(\hat{x}^{\kappa}_{fa}), \\ \hat{y}^{\kappa}_{(f+1)a} &= \mu^y(\hat{y}^{\kappa}_{fa}). \end{aligned} \tag{4.11}$$

The primary architecture of regression network is made in fully connected layers of 100, 80, 60, 40, 40, and 20 neurons with ReLU as activation function for individual $\hat{x}^{\kappa}_{fa}$ and $\hat{y}^{\kappa}_{fa}$ presented at Figure 4.3. MSE loss is used to predict position of nutriments in next frame. Predicting $\hat{y}^{\kappa}_{(f+1)a}$, it requires two inputs $\hat{x}^{\kappa}_{fa}$ and $\hat{y}^{\kappa}_{fa}$ because in the object trajectory, y component has positive and negative value depending on turning point. To detect turning point, the position of component x is necessary to classify the movement of nutriments in correct direction.

Counting Nutriments

Assuming that bottom-left points lb_{f1} as the center, it can be used to define passed line to count the nutriment. Given linear function $y^l(f) = \alpha_f x^l_f + \beta_f$ is created from two points $(\hat{x}^l_{f1}, \hat{y}^l_{f1})$ and $(\hat{x}^l_{f2}, \hat{y}^l_{f2})$. These points is produced from the corner and center of ripple which defined as:

$$\begin{aligned}
\hat{x}^l_{f1} =& \hat{x}^{tl}_{f1}, \\
\hat{y}^l_{f1} =& \hat{y}^{br}_{f1} - \frac{z}{\rho}, \\
\hat{x}^l_{f2} =& \hat{x}^l_{f1} + (\hat{x}^r_{f2} - \hat{x}^r_{f1}), \\
\hat{y}^l_{f2} =& \hat{y}^l_{f1} - (\hat{y}^r_{f1} - \hat{y}^r_{f2}),
\end{aligned} \tag{4.12}$$

where $\rho = 3.6$ is constant value. ρ is used to define the distance between passed line and ripple area. The gradient α_f and coefficient β_f of passed line $y^l(f) = \alpha_f x^l_f + \beta_f$ is calculated as:

$$\begin{aligned}
\alpha_f =& \frac{\hat{y}^l_{f2} - \hat{y}^l_{f1}}{\hat{x}^l_{f2} - \hat{x}^l_{f1}}, \\
\beta_f =& \hat{y}^l_{f1} - \alpha_f \hat{x}^l_{f1}.
\end{aligned} \tag{4.13}$$

4.3 Experiment

In experiment, we describe evaluation approach to precisely presenting quantitative evaluation with various regression models.

4.3.1 Evaluation Approach

Evaluation approach is computed by measuring minimum euclidean distance based on result of regression models with ground truth μ_{gt}. Best regression model μ^* with minimum error distance to the ground truth μ_{gt} is defined as:

$$\mu^* = \arg\min_{\iota}((\mu_{gt} - \sqrt{(\mu^x_\iota(\hat{x}^\kappa_{fa}) - \mu^y_\iota(\hat{y}^\kappa_{fa}))^2})^T(\mu_{gt} - \sqrt{(\mu^x_\iota(\hat{x}^\kappa_{fa}) - \mu^y_\iota(\hat{y}^\kappa_{fa}))^2})), \tag{4.14}$$

where $\iota \in [1,6]$.

4.3.2 Quantitative Evaluation

In this section, we aim to answer two main questions towards understanding our model. Firstly, we compare the performance of different regression models and observe performance of proposed method. In Table 4.1 shows the description of various regression models namely $R1$, $R2$, $R3$, $R4$, $R5$, and $R6$. We train each regression model with same parameter with 10^6

Table 4.1 Different regression models for evaluation process. Each model has different fully connected (fc) layers and different number of neurons.

Methods	fc1	fc2	fc3	fc4	fc5	fc6	fc7	fc8	fc9	fc10
*R*1	100	80	60	40	40	20				
*R*2	200	160	120	80	80	40				
*R*3	100	90	80	70	60	50	40	30	20	10
*R*4	200	180	160	140	120	100	80	60	40	20
*R*5	100	40								
*R*6	200	80								

Table 4.2 Statistical analysis of various regression model *R*1, *R*2, *R*3, *R*4, *R*5, and *R*6 for quantitative evaluation using one-samples T-Test.

Methods	N	Mean (pixels)	Std. Dev. (pixels)	Std. Err. (pixels)	95% Confidence Interval of the Diff.	
					Lower (pixels)	Upper (pixels)
*R*1	50	**10.41**	5.79	0.82	**8.76**	**12.05**
*R*2	50	10.84	5.77	0.82	9.20	12.48
*R*3	50	10.94	**5.74**	**0.81**	9.31	12.58
*R*4	50	10.89	5.81	0.82	9.24	12.54
*R*5	50	10.90	5.88	0.83	9.23	12.57
*R*6	50	10.99	5.82	0.82	9.34	12.64

iterations and 10^{-7} as learning rate. For conclusion, we present the result for each regression model in Table 4.2. By following quantitative evaluation, regression model $R1$ has lowest minimum error distance and we apply this model to be a base of regression model.

4.4 Result

Before we start to explain result of proposed method, we firstly show evaluation result for better understanding the behavior of proposed method. We then compare proposed method and state-of-the-art benchmark methods of tracking algorithm on our datasets and show the computational time to figure out the performance and advantage proposed method comparing state-of-the-art benchmark method.

4.4.1 Evaluation Result

To acknowledge the performance YOLOv4 [17], mAP and loss are used as the training parameter. To train object detection using our datasets, the detail of parameter uses $10k$ iterations with 512×288 pixels for image resizing from 1920×1080 pixels. Figure 4.4 displays training result and reaches 75% of maximum mAP and 0.9 as loss value after two days training using GeForce RTX 2080 Ti. Figure 4.5 show the result of proposed method in sequential frames of or video. By following this graph, number of nutriments can clearly use to identify the behavior of fish feeding activity. The number of nutriment is calculated by cumulative summation of nutriments in 20 frames. We also tested performance regression of proposed method with state-of-the-art tracking methods. Many of state-of-the-art tracking methods use multiple object tracking (MOT) [105, 74, 101, 117, 113, 12, 78]. These methods perform well in six publicly databases on pedestrian detection, MOT and person search provided by [71, 57, 30]. For evaluations, JDE method [105] has been chosen to represent MOT as benchmark method because its method has an accurate prediction based on re-implementation of faster object detection compared with [74, 101, 117, 113, 12, 78]. We also use trajectory mapping [82] as a benchmark method because its method has a good performance in application to aquaculture fish tank. The last method to be a benchmark method is SORT [16]. We add the detection model of trajectory mapping to completely understand performance of tracking method.

In Figure 4.6a, the proposed method is demonstrated to be able to track small nutriment well while trajectory mapping(TM), JDE and SORT with original YOLOv3 and trajectory mapping detection model perform poor (Figure 4.6b, 4.6c, 4.6d, 4.6e) without tracking results of nutriments even trajectory mapping and SORT are able to detect some nutriments and ripple area. In this cases, we assume that the benchmark methods fail to run our datasets because the

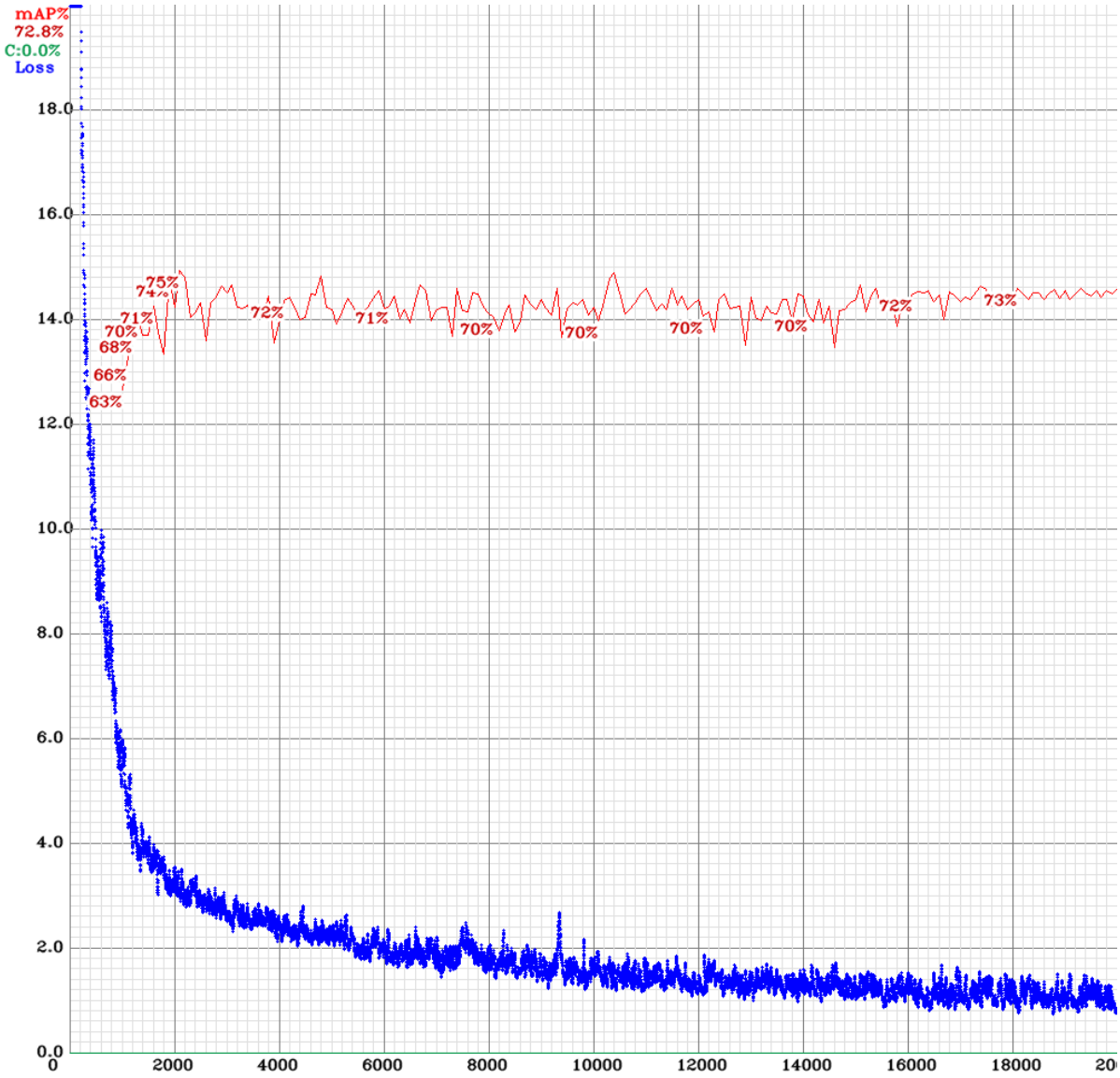


Fig. 4.4 mAP and loss for all iterations during training model using YOLOv4 [17]. After 2k iterations, mAP curve of training model has more stable and loss value has slowly decreased for every iteration.

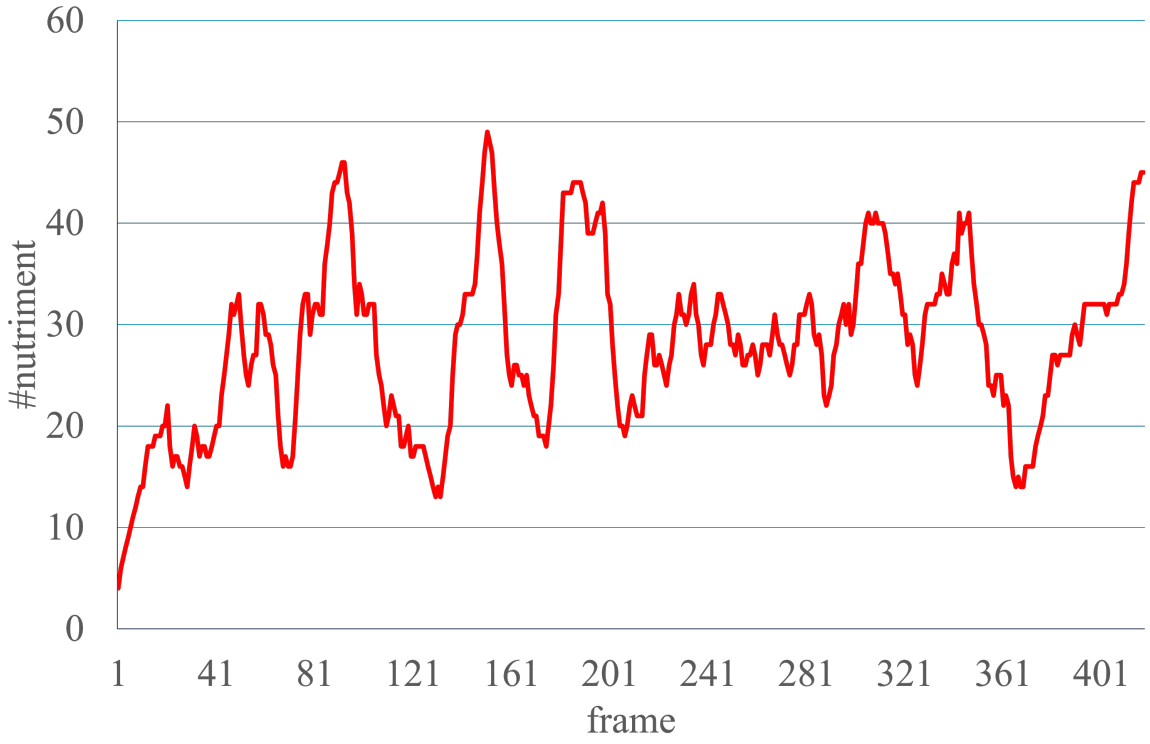


Fig. 4.5 Result of regression and estimate of ripple behavior. This graph shows number of nutriment in each frame. Based on this graph, the process of feeding activity is available.

Table 4.3 Comparison computational time proposed method and benchmark methods.

Methods	N	Mean (fps)	Std. Dev. (fps)	Std. Err. (fps)	95% Confidence Interval of the Diff.	
					Lower (fps)	Upper (fps)
Ours	418	**3.86**	**0.11**	**0.01**	**3.85**	**3.87**
Trajectory Mapping(TM)	418	1.93	0.61	0.03	1.87	1.99
JDE	418	1.87	0.07	0.00	1.86	1.87
YOLOv3(TM) +Sort	418	0.47	-	-	-	-
YOLOv3 +Sort	418	0.45	-	-	-	-

size of nutriment is too small (maximum size is 13×36 pixels) and the speed of nutriment is fast (average nutriment movement from fish feeding machine to ripple area is 23.8 frames).

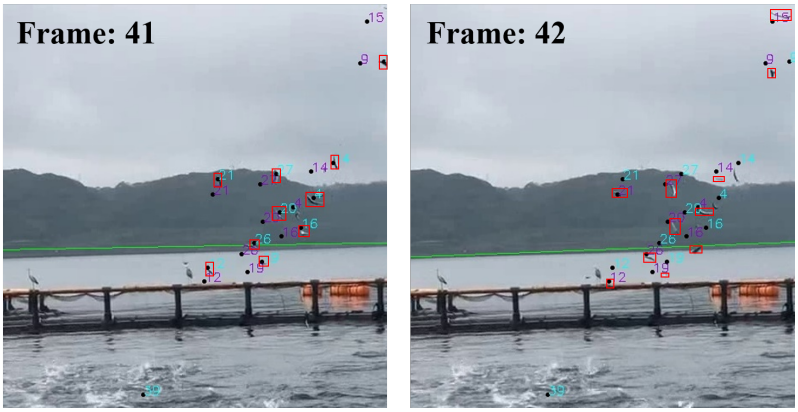
4.4.2 Computational Time

Table 4.3 presents comparison of computation time (in fps) between proposed method and benchmark methods (trajectory mapping, JDE and SORT with original YOLOv3 and trajectory mapping detection model). We reach 3.86 and 0.11 fps for average and standard deviation of computational time, respectively, while trajectory mapping needs 1.93 and 0.61 fps and JDE spends 1.87 and 0.07 fps in which these benchmark methods runs twice slower than proposed method. SORT provides average computational time without information of computational time for individual frame. Computational time for both detection model of YOLOv3 and trajectory mapping model with SORT performs worst comparing with proposed method and other benchmark methods in which it runs 0.45 and 0.47 fps, respectively. By analyzing computational complexity, proposed method is the fastest model with twice for the different speed comparing benchmark methods.

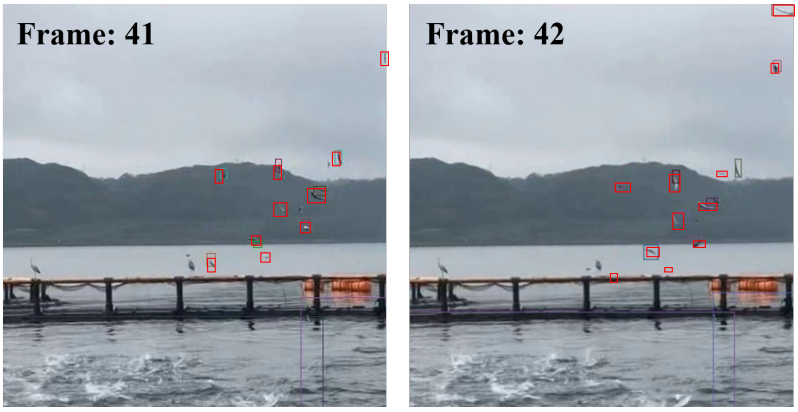
4.5 Conclusion and Discussion

Feature extraction of nutriment is important to optimize the amount of nutriment giving to the fish. Recent studies have shown that it is possible to track all detected objects in sequence frames on video. However, there is no agreement to track multiple small and dense nutriments and also to detect ripple activity area in the video. In this chapter, feature extraction of nutriment has been presented and demonstrated to be promising for optimizing fish feeding process to give an optimal rate of both costs and profits. We expect proposed method to open the door for

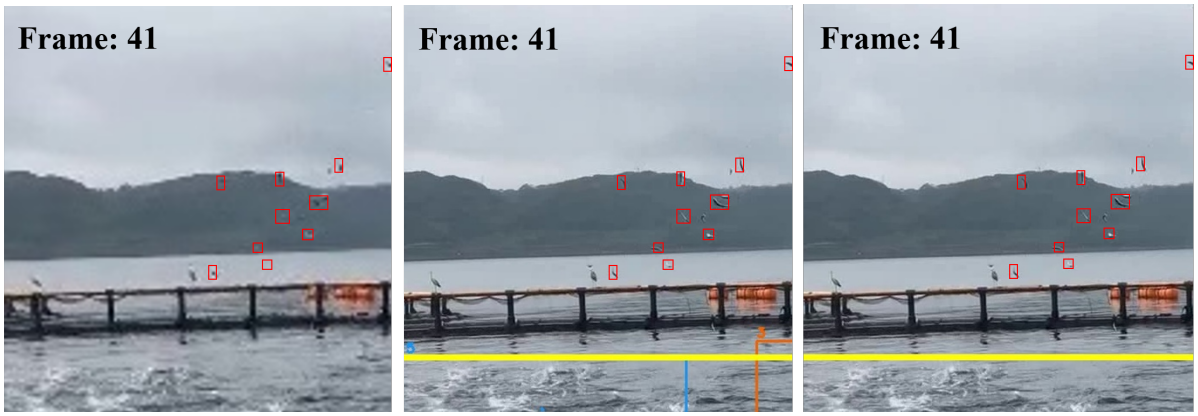
future work and to go beyond for developing a large community-generated database and focus on integrating with the sensors to give more accurate and robust results.



(a) Proposed Method



(b) Trajectory Mapping (TM)



(c) JDE (d) YOLOv3(TM) + SORT (e) YOLOv3 + SORT

Fig. 4.6 Observation of proposed method shown in (a) and the results of benchmark methods shown in (b), (c), (d) and (e). Red box in each image represents ground truth of nutriment. In proposed method, there are two different colors to represent $\hat{x}^{\kappa}_{fa}$ as blue light color and $\hat{x}^{\kappa}_{(f+1)a}$ as purple. We can see that red box and purple points in proposed method show precisely tracked result and it proves that proposed method predict very well while benchmark methods perform poor without tracking results of nutriments even trajectory mapping SORT are able to detect some nutriments.

Chapter 5

Variance of VGG Texture

5.1 Introduction

Raising fish utilization to complete the demand of human and animal resourcing in worldwide drives researchers exploring and improving technology in aquaculture industry [98, 5, 49]. Based on global aquaculture production statistic database [FAO], the proportion of aquatic animals farmed is 55.1 million tonnes in 2009 and significantly increasing to 82.1 million tonnes in 2018. This trend indicates that aquaculture industry has to create new technique to enlarge economic scale with reducing production cost and increasing production efficiency.

In this chapter, we present an automatic controlling fish feeding machine using textural feature to estimate ripple behavior. Our method is constructed by object detection of ripple. Estimating ripple behavior is determined by textural feature with ripple detection area as input image to compute global variance of VGG network [91]. Textural feature of ripple behavior can be useful to improve the production profit in fish farms by controlling the amount of nutriment in optimal rate and to optimize the use of fish feeding machine. The datasets are extracted from a video camera which is set down above vessel with a highly distraction of ocean wave and innumerable small and dense nutriments. After that, the camera captures above sea surface and the videos transfer to a computer for further analysis.

5.2 Proposed Method

Our formulation is based on textural feature of ripple behavior as input image to compute global variance of VGG network. In order to provide some background and formally introduce our approach, we start by providing diagram of textural feature of ripple behavior approach. We then explain how the proposed method works to real environment. The proposed method

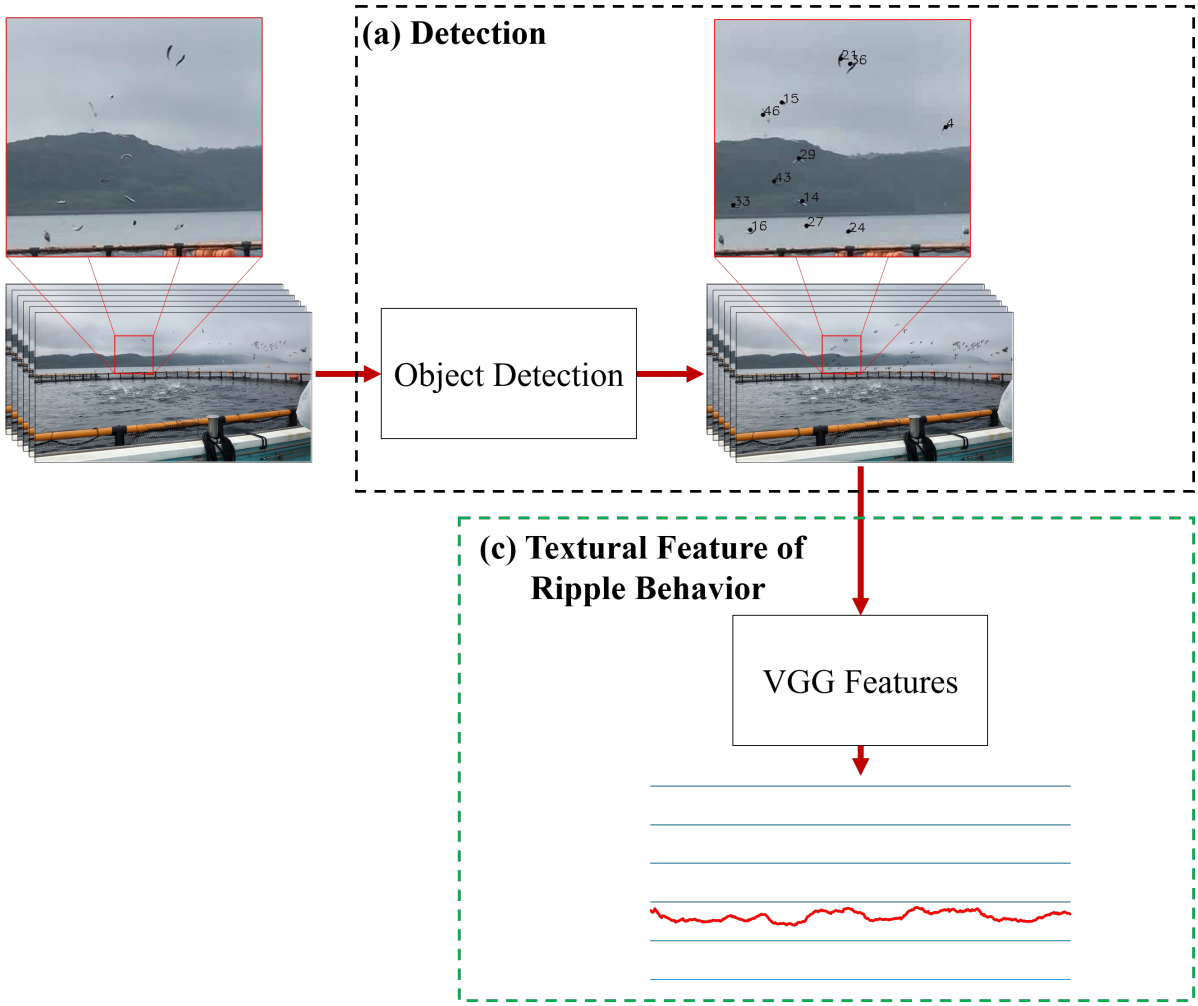


Fig. 5.1 Diagram of the proposed method. (a) The input video is received and applied object detection using YOLOv4 [17]. (b) Textural feature of ripple behavior uses global variance of VGG model to represent estimation of ripple behavior for each frame on input video.

contains three parts: detection, regression and textural feature of ripple behavior which are shown in Figure 5.1 where the detection result is represented as black color value and predicted result of regression is shown by red color value. Regression contains both transformation and inverse transformation, regression approach and counting nutriments while textural feature of ripple behavior uses VGG network as extracted features to compute global variance of VGG network.

5.2.1 Textural Feature of Ripple Behavior

Ripple activity level is determined as human perception for the size and number of ripple appearing from fish feeding process. Human perception cannot continuously be used to adjust fish feeding machine because its perception has no standard for validation or subjective methods. By following that, we propose robust method to estimate ripple behavior using VGG as base model for feature extraction which is presented in Figure 5.2. This idea is adopted from full reference image quality assessment (FR-IQA). FR-IQA is assessment the quality of of test images comparing with reference image for standard quality. The difference of FR-IQA and our approach is that FR-IQA requires reference and test image while our approach only needs single images to define ripple activity level.

VGG Feature

Given cropped image d_f from point tl_{f1} to br_{f2} in original image as the input image of VGG network, VGG feature of convolutional layers is calculated by:

$$f(d_f) = \{\hat{d}^{(i)}_{f(j)}; i = 1, \cdots, \omega; j = 1, \cdots, \phi_i\}, \tag{5.1}$$

where $\omega = 5$ represents the number of VGG convolution layers. ϕ_i is the number of feature maps in i^{th} VGG convolution layers. Variance $\sigma^{(i)}_f$ of feature maps in i^{th} VGG convolution layers is presented as:

$$\sigma^{(i)}_f = \sqrt{\frac{\Sigma(\sigma^{(i)}_{f(j)} - \frac{\Sigma\sigma^{(i)}_{f(j)}}{\phi_i})^2}{\phi_i}}, \tag{5.2}$$

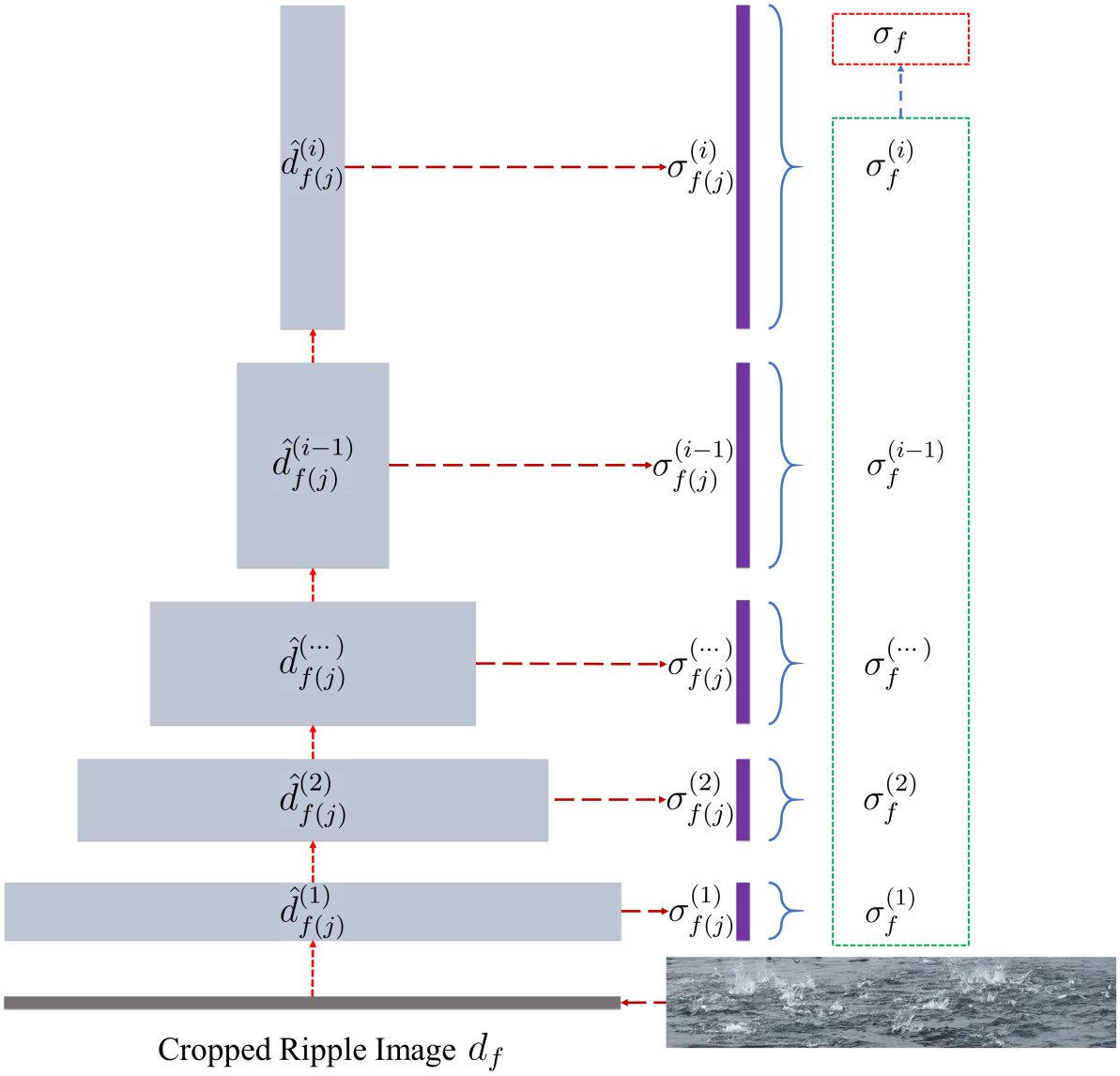


Fig. 5.2 VGG architecture for calculating global variance σ_f to estimate level of ripple activity.

(a) Fish tank area without ripple behavior

(b) Fish tank area with ripple behavior

Fig. 5.3 In this observation, we estimate ripple behavior using image with and without ripple area to compromise global variance of VGG network.

where $\sigma^{(i)}_{f(j)}$ is variance of each feature maps $\hat{d}^{(i)}_{f(j)}$ in VGG convolution layers. Global variance σ_f is computed by:

$$\sigma_f = \sqrt{\frac{\sum(\sigma^{(i)}_f - \frac{\sum \sigma^{(i)}_f}{\omega})^2}{\omega}}. \tag{5.3}$$

5.3 Experiment

We then describe evaluation approach to precisely presenting quantitative evaluation with various regression models.

5.3.1 Evaluation Approach

To observe estimation of ripple behavior, we use different image to represent performance of global variance of VGG network which is shown in Figure 5.3. For fair comparison, we use the image with same dimension which is 843×162 pixels. In conclusion, image without ripple in Figure 5.3a archives 0.029 while image with ripple in Figure 5.3b reaches 0.3037.

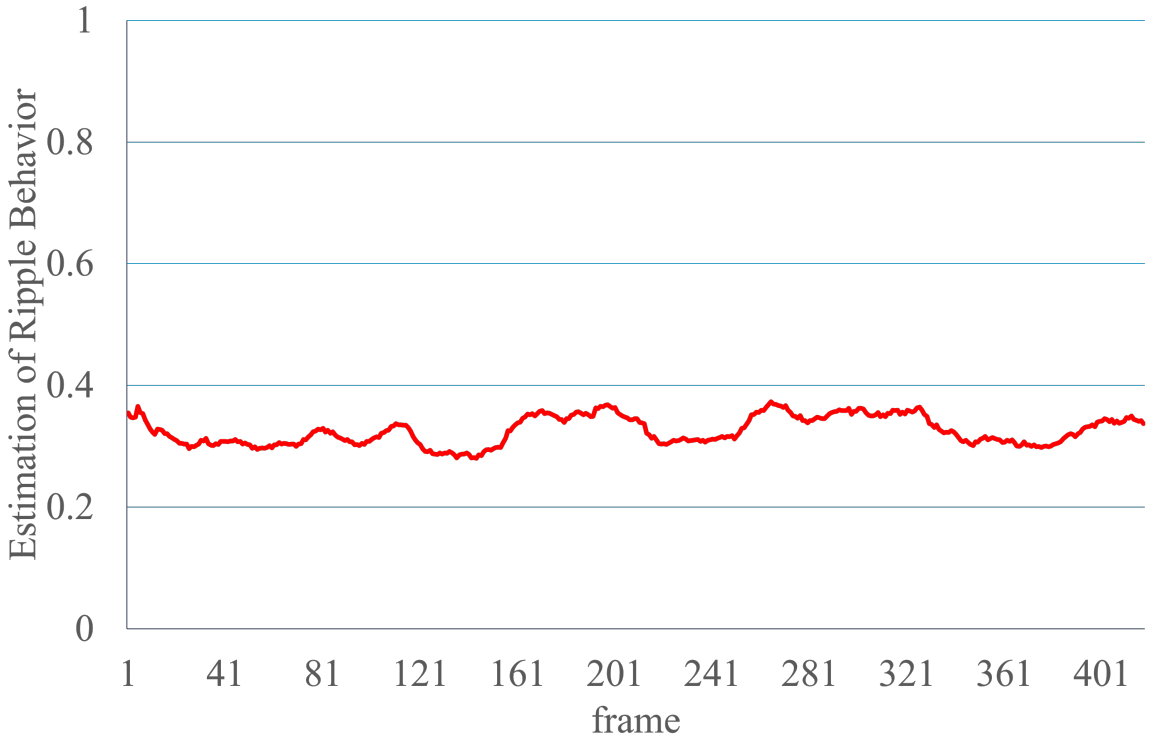


Fig. 5.4 Estimation of ripple behavior. This graph represents ripple behavior. Based on this graph, the process of feeding activity is available.

5.3.2 Result

Figure 5.4 show the result of proposed method in sequential frames of or video. By following this graph, estimation of ripple behaviors can clearly use to identify the behavior of fish feeding activity. Estimation of ripple behavior is computed by cumulative average of global variance in 20 frames.

5.4 Conclusion and Discussion

Feature extraction is important to optimize the amount of nutriment giving to the fish. Feature extraction of ripple behavior has presented an agreement to complement nutriment feature. This feature has been presented and demonstrated to be promising for optimizing fish feeding process to give an optimal rate of both costs and profits. We have demonstrated an area with and without ripple activity which has a big difference gap and the proposed method consistently performs well on the video contains small and dense nutriments. We expect proposed method to open the door for future work and to go beyond for developing a large community-generated database and focus on integrating with the sensors to give more accurate and robust results.

Chapter 6

Image Quality Assessment

6.1 Introduction

In recent years, camera technology affects the explosive growth of image sharing on social networking. Massive number of images play important role on image compression [21, 66, 4] and restoration [68, 70, 115] which will affect loss of visual information. In consequence, flexibility of image quality metric is the most influential aspect to measure image quality from human perception. Although human assumption can be standard of subjective evaluation, cost and time consuming are current problem in real application. So that, estimating of visual quality model has more attractive research topic to generalize ability of image quality model. To develop objective image quality assessment (IQA), human perception is required to be standard quality assessment to produce imitating quality assessment model which is more reliable solution to identify quality of images [52]. Visualizing image quality is to assess subjective calculation of human perception to be quantity measurement as mean opinion scores (MOS). MOS is standard quantity assessment referred to the perceptual image quality metric, which is important aspect for evaluation criteria on imaging technologies.

In real applications, human assumption can easily obtain quality of image by collecting information of object poses, image foreground position and blurry image background. Therefore, collecting this information has crucial component for learning model with different features. Separating these features can increase learning accuracy of NR-IQA method. From this motivation, we propose a new novel NR-IQA method based on separating convolutional features which make powerful model with integration of different convolutional kernel sizes and demonstrate its strength using quantitative evaluations. Quantitative evaluations are conducted based on cross database evaluation and show that proposed method is an effective way to increase learning accuracy and can be widely applied to various models. To summarize our work, we present several **contributions**:

- We propose a new novel feature processing of NR-IQA method based on different convolutional features which can exploit performance of learning model.
- We propose different ways to improve learning accuracy without changing training parameter by using separation foreground and background features.
- We have done experiments with crossing evaluation from four public IQA databases which have artificial and authentic distortions and show significantly improvement results of proposed method comparing state-of-the-art benchmark methods.

6.2 Related Work

In this section, we categorize NR-IQA methods according to network architecture: conventional and deep learning approaches further used to evaluate performance of proposed method.

6.2.1 NR-IQA Based on Conventional Approach

In the beginning of NR-IQA methods, feature extractor was used to transform single image into a representative feature vector using support vector regression to estimate the MOS value. Several conventional NR-IQA methods impersonated the natural scene statistics model [94] to capture image behaviors. DIIVINE [73] proposed extracting scene statistics from a distorted natural image with quality evaluation for the specific distortion on the image using two-step frameworks. Another type for conventional NR-IQA methods was CORNIA [112] which used code-book for local features encoding and feature pooling. This method based on an unsupervised feature learning approach assigned raw-image-patches using max pooling to encode the dictionary for computing efficiency. Another effective and efficient method, BRISQUE, was introduced by [72]. This method adopted pair-wise products of locally normalized intensity coefficient. Parameter of this model quantified the natural losses of the image due to availability of distortion. To calculate mean of each cluster, HOSA was proposed by [109] which computed the differences from high order statistics including skewness, mean, and variance between local features of image and corresponding clusters. This method was also able to map the relationship between small code-book and features encoding more thoroughly and its method had been proven to improve the quality prediction performance.

6.2.2 NR-IQA Based on Deep learning Approach

Instead of carefully designing conventional feature extraction, NR-IQA based on deep learning approach was able to increase learning accuracy for automatically discovering model interpreta-

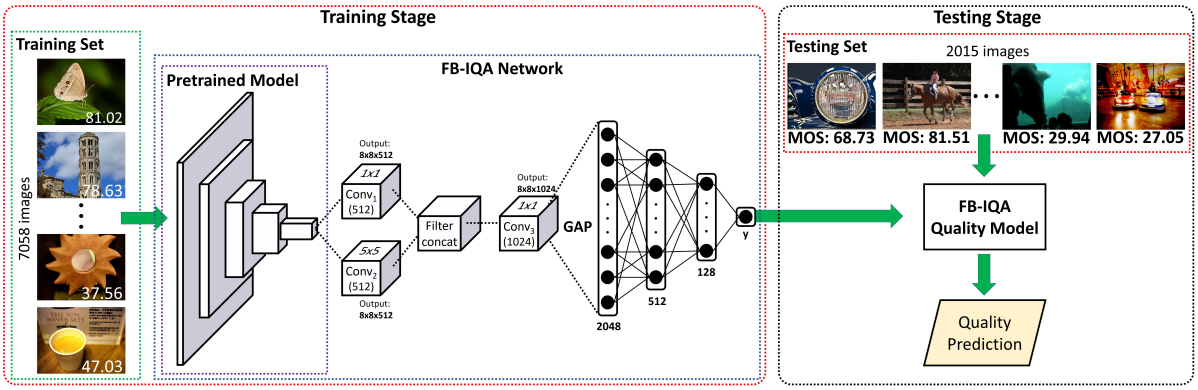


Fig. 6.1 The overview of Foreground-Background-IQA (FB-IQA) architecture.Chapter6/ Feature extraction from pretrained model is further used to analyze separating foreground and background features with different convolutional kernel sizes. FB-IQA architecture has three convolutional boxes namely $Conv_1$, $Conv_2$, $Conv_3$ and one filter concatenation (filter concat). Training and testing sets are randomly divided into 70% (7058 images) and 20% (2015 images) using KonIQ-10k database [63].

tion to estimate quality scores from images. Although few availability existing IQA databases was unsuitable to apply end-to-end learning, several NR-IQA based deep learning approaches used pretrained model of DCNNs as feature extractors from images for further assessing image quality based on deep learning approach. Recently, DeepRN [104] presented improving prediction performance using large image size rather than re-sized input images. Derived from DeepRN, Koncept was proposed by [46]. This method was trained on a higher resolution image than DeepRN [104] models. Architecture of this method was created by feature extraction from the Inception-ResNet-v2 [99] followed by global average pooling (GAP) and fully connected (fc) layers to predict the value of MOS. A new training strategy of NR-IQA method based on meta-learning was introduced by [122]. This method had the ability to quickly learn with small training sample and unknown distortions. To determine two-level gradient descent approach, this method used specific distortion images to learn a meta-model.

6.3 Our Approach

Fig. 6.1 shows our proposed method architecture namely Foreground-Background-IQA (FB-IQA) which separates foreground and background features using different convolutional features. Overall, the system architecture inputs a color image and predicts its quality prediction score. Therefore, training and testing sets are randomly divided into 70% (7058 images) and 20% (2015 images) using KonIQ-10k database [63]. To estimate image quality prediction,

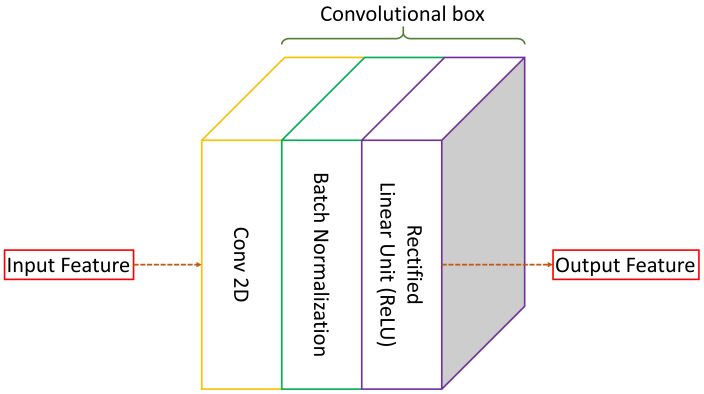


Fig. 6.2 There are three layers inside of convolutional (Conv) box: convolution 2D (Conv 2D), batch normalization, and rectified linear unit (ReLU).

proposed method requires training set and corresponding empirical score of MOS. The detail of architecture of FB-IQA and data transformation are outlined in below.

6.3.1 FB-IQA Architecture

Our proposed system relies on pretrained model of Inception-Resnet-v2 architecture [99] as a base model which are interesting result on image classification [27]. Feature extraction from pretrained model is further used to analyze separating features with different convolutional features. Detail information inside of each convolutional box is presented in Fig. 6.2. Each convolutional box has three layers: convolution 2D, batch normalization, and rectified linear unit (ReLU).

Given an input feature $X(B,C_x)$ of convolution 2D, the output feature after convolution 2D $\hat{X}(B,C_{\hat{x}})$ is defined by:

$$\hat{X}(B_i,C_{\hat{x}_j}) = \delta_{C_{\hat{x}_j}} + \sum_{k=0}^{C_x-1} \varphi_{C_{\hat{x}_j}}(k) \star X(B_i,k), \tag{6.1}$$

where δ and φ are the learnable bias and weight to reflecting input convolutional layer X to output convolutional layer $\hat{X}$. $\star$ is the valid 2D cross-correlation operator while B and C are a batch size and a number of channels, respectively. In our proposed system, the batch size B is set to 10. To define learnable weight φ, this function is randomly sampled from $-\varphi$ to φ which is defined as follow:

$$\varphi_{C_{\hat{x}_j}}(k) = \sqrt{\frac{1}{k * \prod_{i=0}^{1} \xi_{C_{\hat{x}_j}}^{i}}}, \tag{6.2}$$

where $\xi_{C_{\hat{x}_j}}$ is kernel size of convolutional layer $\hat{X}$. Negative learnable weight $-\varphi$ can reduce the value of convolutional output $\hat{X}$ while positive learnable weight φ shows the important value of convolutional output $\hat{X}$. In Fig. 6.1, our network has three convolutional boxes namely Conv_1, Conv_2 and Conv_3. The kernel size ξ for each convolutional box is empirically defined as 1×1 for Conv_1, Conv_3 and 5×5 for Conv_2.

After computing convolution 2D, batch normalization is applied to speed up convergence as described in [48]. Given a layer with j-dimensional output feature after convolution 2D, $\hat{X} = (\hat{x}^1, \hat{x}^2, \cdots, \hat{x}^j)$ will be normalized for each dimension to be $Y = (y^1, y^2, \cdots, y^j)$ presented by:

$$y^j = \frac{\hat{x}^j - \mu(\hat{x}^j)}{\sqrt{\sigma(\hat{x}^j) + \varepsilon}} * \gamma + \beta, \tag{6.3}$$

where μ and σ are mean and variance of $\hat{x}^j$ and will be optimized during training process. γ and β are learnable of vectors $\hat{X}$. In our experiment, we use default value of γ and β which set to 1 and 0, respectively. During evaluation, μ and σ are used for normalization with a default value for momentum set to 0.1. ε is learning rate applied to control numerical stability and error handle of the variance σ.

After batch normalization, we apply ReLU [75] as activation function to speed up the convergence of loss value during training process [54] calculated as follows:

$$\text{ReLU}(Y) = \max(0, Y). \tag{6.4}$$

Beside of three convolutional boxes, our network also has one filter concatenation (filter concat). For each convolutional box, we set up $C_{\hat{x}}^1 = 512$, $C_{\hat{x}}^2 = 512$, and $C_{\hat{x}}^3 = 1024$ for Conv_1, Conv_2, and Conv_3, respectively. Given $C_{\hat{x}}^1$, $C_{\hat{x}}^2$ as input of filter concatenation, the number of channel after filter concatenation C_ω is defined by:

$$C_\omega = C_{\hat{x}}^1 + C_{\hat{x}}^2. \tag{6.5}$$

Our proposed model involves two elements: convolutional layers with GAP followed by fc layers. GAP is a pooling layer with adaptive averaging calculation on entire input signal while fc layers are sequential layers where all outputs from one layer are connected to be the inputs in the next layer. Fc layers can easily produce over-fitting during training process. To avoid it, drop-out [95] function is applied to solve over-fitting between training and validating images. This function randomly applies zero values with probability λ using a Bernoulli distribution for some input elements of fc layer. For each channel, the values will be forced to independently be zero for each forward call. As described in [44], this function has effectively proven to regularize and prevent the ability for neuron adaption. In our experiments, we use four fc layers, namely fc_1, fc_2, fc_3, and fc_4 with 2048, 512, 128, 1 of neurons. For each fc layer fc_1, fc_2, and fc_3, we apply different probability of drop-out $\lambda_1 = 0.25$, $\lambda_2 = 0.25$, and $\lambda_3 = 0.4$, respectively.

6.3.2 Data Transformation

Image transformation is applied during training process to build more complex transformation pipeline. In our network, we have three different image transformations namely T_1, T_2, and T_3. These transformations are used for manipulating input image I in sequentially processing. T_1 and T_2 use random horizontal flip with probability 0.5 while T_3 applies random rotation from $-90°$ to $90°$ and random scaling from 0.9 to 1.1. Our transformation only uses different image perspectives without changing intensity and color values because in our assumption, changing intensity and color values will affect the image quality itself. Our optimization function uses Adam [53] as gradient-based optimization of stochastic objective functions in which the detail of optimization process is further presented in next section.

6.3.3 Optimization Process

Based on Algorithm 3, our goal is that we input images from testing set J into FB-IQA network to generate the predicted quality score q defined as follow:

$$q = f(J; \chi_{best}), \tag{6.6}$$

where χ_{best} denotes the best network parameter of FB-IQA. Given training set images I to be mini-batch I_{K_τ}, we use mean square error to minimize loss function between the predicted and ground-truth quality scores of mini-batch I_{K_τ} taking the following form:

Algorithm 3: FB-IQA algorithm

Input : Images from training set: I, images from testing set: J, initial model parameters χ_0, ground-truth MOS p.
Output : Predicted quality score q for J.

Randomly initialize model parameters χ_0;
Initialize $v_{best} \leftarrow 9999$ and $\chi_{best} \leftarrow \chi_0$;
for $\tau \leftarrow 1$ *to* 3 **do**
  for *iteration* $= 1, 2, \ldots$ **do**
    Mini-batch I_{K_τ} in training set I;
    for $k \leftarrow 0$ *to* K_τ **do**
      Compute $v_k = (f(\upsilon_{K_\tau}; \chi_k) - p)^2$;
      if $v_k < v_{best}$ **then**
        Update $v_{best} \leftarrow v_k$ and $\chi_{best} \leftarrow \chi_k$;
      end
      Compute $\chi^{k+1} = Adam(v_k, \chi_k)$;
    end
  end
end
Input J into the quality model χ_{best};
Return q.

$$v_k = (f(I_{K_\tau}; \chi_k) - p)^2, \tag{6.7}$$

where χ_0 is randomly defined as the initialized network parameters. p denotes the ground-truth quality score of mini-batch I_{K_τ}. Therefore, the first-order gradients of loss function $v(k)$ relates to all model parameters defined as:

$$g_k = \nabla v_k(f(\chi)). \tag{6.8}$$

Then, we update the model parameters for $K_\tau \in \{1, 2, \cdots, k\}$ steps using the Adam [53] optimizer which is shown as:

$$\text{Adam}(v_k, \chi_k) : \chi^{k+1} \leftarrow \chi^k - \alpha_\tau \frac{m_{k+1}}{\sqrt{n_{k+1}} + \varepsilon}, \tag{6.9}$$

where ε is learning rate applied to control numerical stability and error handle of $\sqrt{n_{k+1}}$ with $1e-8$. α_τ is the inner learning rate with $\tau \in \{1,2,3\}$. In our experiment, we define α_1, α_2, and α_3 with $1e-4$, $1e-4$, and $1e-5$, respectively. m_{k+1} and n_{k+1} denote the first moment and second raw moment of gradients, which are formulated as:

$$m_{k+1} = (1-\delta_1^{k+1}) \sum_{i=0}^{k} \delta_1^{k-i} \cdot g_{i+1}, \tag{6.10}$$

$$n_{k+1} = (1-\delta_2^{k+1}) \sum_{i=0}^{k} \delta_2^{k-i} \cdot g_{i+1}^2, \tag{6.11}$$

where δ_1 and δ_2 are the exponential decay rates of m_{k+1} and n_{k+1}, respectively.

6.4 Optimization Process

Algorithm 4: FB-IQA algorithm

Input : Training set of images: I, testing set of images: J, initial model parameters χ_0, ground-truth MOS p.
Output : Predicted quality score q for J.

Randomly initialize model parameters χ_0;
Initialize $v_{best} \leftarrow 9999$ and $\chi_{best} \leftarrow \chi_0$;
for $\tau \leftarrow 1$ *to* 3 **do**
  for *iteration* $= 1,2,\ldots$ **do**
    Mini-batch I_{K_τ} in training set I;
    for $k \leftarrow 0$ *to* K_τ **do**
      Compute $v_k = (f(\upsilon_{K_\tau}; \chi_k) - p)^2$;
      if $v_k < v_{best}$ **then**
        Update $v_{best} \leftarrow v_k$ and $\chi_{best} \leftarrow \chi_k$;
      end
      Compute $\chi^{k+1} = Adam(v_k, \chi_k)$;
    end
  end
end
Input J into the quality model χ_{best};
Return q.

Based on Algorithm 4, our goal is to input testing set of images J into FB-IQA network to generate the predicted quality score q defined as follow:

$$q = f(J; \chi_{best}), \tag{6.12}$$

where χ_{best} denotes the best network parameter of FB-IQA. Given training set images I to be mini-batch I_{K_τ}, we use mean square error to minimize loss function between the predicted and ground-truth quality scores of mini-batch I_{K_τ} taking the following form:

$$v_k = \frac{\sum(f(I_{K_\tau}; \chi_k) - p)^2}{\#p}, \tag{6.13}$$

where p denotes the ground-truth quality score of training set of images I. Therefore, the first-order gradients of loss function $v(k)$ relates to all model parameters defined as:

$$g_k = \nabla v_k(f(\chi)). \tag{6.14}$$

Then, we update the model parameters for $K_\tau \in \{1, 2, \cdots, k\}$ steps using the Adam [53] optimizer which is shown as:

$$\text{Adam}(v_k, \chi_k) : \chi^{k+1} \leftarrow \chi^k - \alpha_\tau \frac{m_{k+1}}{\sqrt{n_{k+1}} + \varepsilon}, \tag{6.15}$$

where ε is learning rate applied to control numerical stability and error handle of $\sqrt{n_{k+1}}$ with $1e-8$. α_τ is the inner learning rate with $\tau \in \{1, 2, 3\}$. In our experiment, we define α_1, α_2, and α_3 with $1e-4$, $1e-4$, and $1e-5$, respectively. m_{k+1} and n_{k+1} denote the first moment and second raw moment of gradients, which are formulated as:

$$m_{k+1} = (1 - \delta_1^{k+1}) \sum_{i=0}^{k} \delta_1^{k-i} \cdot g_{i+1}, \tag{6.16}$$

$$n_{k+1} = (1 - \delta_2^{k+1}) \sum_{i=0}^{k} \delta_2^{k-i} \cdot g_{i+1}^2, \tag{6.17}$$

Database	Distortion type	No. of Images	Score type	Score range
LIVE [89]	artificial	987	DMOS	[0, 100]
CSIQ [55]	artificial	866	DMOS	[0, 1]
LIVE-itW [34]	authentic	1,162	MOS	[0, 100]
KonIQ-10k [63]	authentic	10,073	MOS	[0, 100]

Table 6.1 Comparison of subjective evaluation IQA databases from human assumption. MOS represents subjective human perception to images while DMOS is different quality scales from reference image.

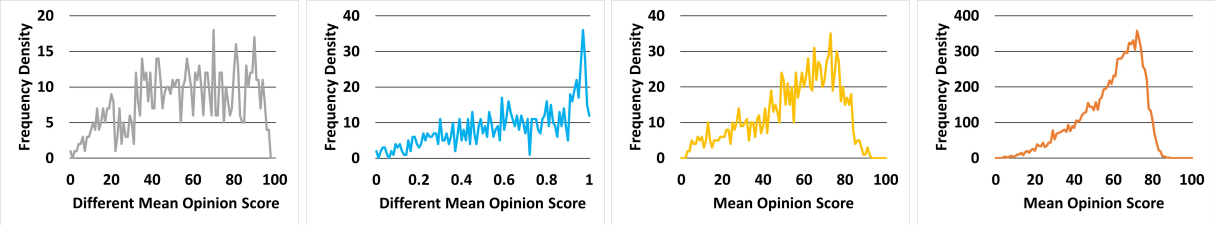
where δ_1 and δ_2 are the exponential decay rates of m_{k+1} and n_{k+1}, respectively.

6.5 Experiments

In this section, we firstly explain the details of databases used to evaluate learning accuracy of proposed and state-of-the-art benchmark methods. After that, evaluation approach applies four different measurements: mean absolute error (MAE), root mean square error (RMSE), pearson's linear correlation coefficient (PLCC), and spearman's rank order correlation coefficient (SROCC) as evaluation criteria. Ablation study presents further exploring the effectiveness of our method comparing with base model. In last, we compare performance of proposed method and state-of-the-art benchmark methods to show that proposed method has superior result among them. For fair comparison, we train and test proposed and benchmark methods for each database and also evaluate their performance with crossing database evaluation.

6.5.1 Databases

We evaluate the performance of proposed method with two types of databases: artificially and authentically distorted IQA databases. Artificially distorted is the distortion type created by adding specific distortion such as blurring [60] to the reference image while authentically distorted is naturally distortion which is no reference image to create it. The detail of information for each database is listed in Table 6.1. In our application, we use two artificially distorted databases (LIVE [89] and CSIQ [55]) and two authentically distorted databases: LIVE in the Wild (LIVE-itW) [34] and KonIQ-10k [63]. In Fig. 6.3, we also present the distribution for each database based on different mean opinion scores (DMOS) or MOS which have various distributions for each database. DMOS is calculated using different quality scales from reference image as standard of perfect quality.



(a) LIVE database [89]. (b) CSIQ database [55]. (c) LIVE-itW database [34]. (d) KonIQ-10k database [63].

Fig. 6.3 The distribution of DMOS and MOS in various score ranges. The quality scale of LIVE [89], LIVE-itW [34], and KonIQ-10k [63] databases is started from 0 (worst) to 100 (best). CSIQ database [55] has the quality scale started from 0 (worst) to 1 (best).

Artificially distorted IQA databases

Most of artificially distorted databases such as LIVE [89] and CSIQ [55] are evaluated by the form of DMOS. LIVE [89] database has six subsets with 987 images. Reference image contains 29 images where each image is chosen with high quality and resolution color images and it is collected from photographic CD-ROMs and internet. These images have several objects including people faces, animals, wide-angle and closeup shots, hand-crafted objects, and nature scenes with different foreground and background poses and appearances. To generalize selection method, these images are chosen without any specific object of interest. User participation evaluates these distorted images based on reference images as the quality scores which have a range from 0 to 100. Higher score represents higher the quality of image and closely to the reference images. Another artificially distorted database, CSIQ [55], consists of 30 reference images. Each reference image applies one of six distortion types. For each distortion, it contains from four to five different distortion levels. These images are evaluated by subjective calculation of human assumption using a linear displacement. The observer perceives side-by-side of all four calibrated LCD monitors with equal viewing distance. From 35 different observers, they create 5000 subjective ratings which also have a range from 0 to 1. A larger value denotes greater visual distortion compared to the corresponding reference image.

Authentically distorted IQA databases

Authentically distorted IQA databases such as LIVE-itW [34] and KonIQ-10k [63] are generally evaluated by the MOS value. LIVE-itW [34] database is taken from mobile camera which has naturally distortion types such as JPEG compression, underexposure, overexposure, motion blur, low resolution image, and noise. This database contains $1,162$ images where the quality scores of these images are obtained by the experiment from crowd-sourcing and the interval

range of quality score is started from 0 to 100. Higher score represents higher the quality of image. A large-scale authentically distorted IQA database, KonIQ-10k, was introduced by [63] which consists of $10,073$ images. These images are selected from a massive public multimedia database and randomly chosen from $9,974,030$ image records. This database is computed by four types of measurements such as sharpness, contrast, colorfulness, and brightness which have the most correlated with human perception. The quality score of each image is computed by averaging of five-point ratings between 120 workers and scoring has the range from 0 to 100. Higher score also indicates higher the quality of image.

6.5.2 Evaluation Approach

In our experiment, we use four different types of measurements such as MAE, RMSE, PLCC, and SROCC to evaluate performance of proposed method and state-of-the-art benchmark methods. MAE and RMSE are generally used to evaluate performance between predicted and actual values. MAE is used to measure errors between predicted and actual values expressing the same phenomenon. Given N as a number of testing images, MAE is computed by:

$$\text{MAE} = \frac{\sum_{n=1}^{N} |p_n - q_n|}{N}, \tag{6.18}$$

where p_n and q_n are the ground-truth and predicted MOS of n-th image. On the other hand, RMSE computes square root of mean error square which is formulated as:

$$\text{RMSE} = \sqrt{\frac{\sum_{n=1}^{N} (p_n - q_n)^2}{N}}, \tag{6.19}$$

Lower value of MAE and RMSE represents better correlation between ground-truth and predicted MOS score. RMSE is a reliable measurement to compare prediction errors of model configurations for a particular variable and it is not between variables. To represent correlation between variables, we use PLCC and SROCC as indicator. PLCC is used to measure the strength of a linear association between two different variables denoted as:

$$\text{PLCC} = \frac{\sum_{n=1}^{N} (p_n - \mu_p)(q_n - \mu_q)}{\sqrt{\sum_{n=1}^{N} (p_n - \mu_p)^2 \sum_{n=1}^{N} (q_n - \mu_q)^2}}, \tag{6.20}$$

Pretrained Model	Methods	MAE ↓	RMSE ↓	PLCC ↑	SROCC ↑
Inception-v4 [99]	Baseline	4.523	5.908	0.924	0.906
	FB-IQA	**4.348**	**5.763**	**0.928**	**0.911**
Inception-Resnet-v2 [99]	Baseline	4.558	5.974	0.922	0.903
	FB-IQA	**4.399**	**5.748**	**0.931**	**0.914**

Table 6.2 Ablation study results (MAE, RMSE, PLCC and SROCC) on KonIQ-10k [63] database with different pretrained models: Inception-v4 [99] and Inception-Resnet-v2 [99]. For comparison, the best results on all measures are shown in boldfaced.

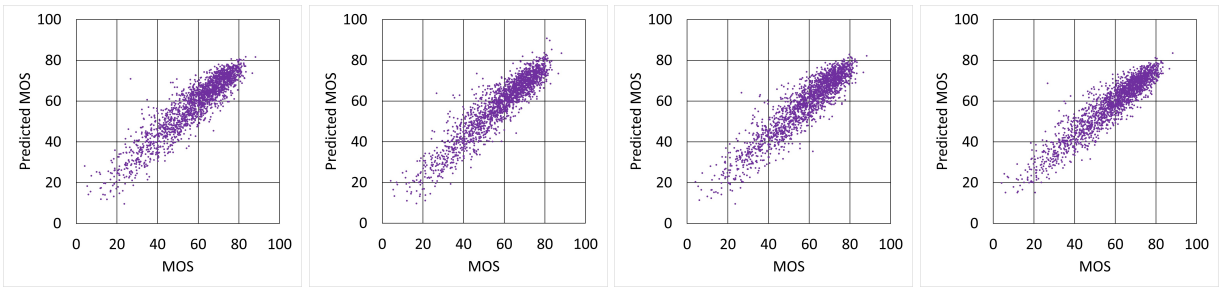
where μ_p and μ_q are global average of ground-truth and predicted MOS, respectively. Let d_n defines difference between the two ranks of each p_n and q_n on n-th image. SROCC can be defined as:

$$\text{SROCC} = \frac{6\sum_{n=1}^{N} d_n^2}{N(N^2-1)}. \tag{6.21}$$

SROCC assesses monotonic relationships whether correlation between these variables is linear or nonlinear. Both PLCC, and SROCC have dynamic range starting from -1 to 1. Positive and negative value represent positive and negative correlation between ground-truth and predicted MOS score, respectively.

6.5.3 Ablation Study

To investigate the effectiveness of FB-IQA architecture, we conduct ablation studies in this experiment. Firstly, we remove three convolutional layers Conv_1, Conv_2, and Conv_3 and filter concatenation in our network (called Baseline). After that, we train and test Baseline and FB-IQA models with the same data transformation. To prove that FB-IQA is effective way, we also use different pretrained models (Inception-v4 [99] and Inception-Resnet-v2 [99]). The database used in this experiment is KonIQ-10k [63] database which will be randomly separated into 70%, 20%, 10% of total images as training, testing, and validating images. The results of all testing images are presented in Table 6.2. Based on this table, it proves that FB-IQA has obvious result compared to Baseline by a large margin on all measures. When we use Inception-Resnet-v2 [99] as pretrained model, FB-IQA reachs 4.399, 5.748, 0.931, and 0.914 with 0.159, 0.226, 0.009 and 0.011 as positive margin from Baseline for MAE,



(a) Inception-v4 [99]+Baseline. (b) Inception-v4 [99]+FB-IQA. (c) Inception-Resnet-v2 [99]+Baseline. (d) Inception-Resnet-v2 [99]+FB-IQA.

Fig. 6.4 Scatter plots of predicted MOS by (a) Baseline and (b) FB-IQA versus ground truth MOS using Inception-v4 [99]. Scatter plots are also used to show predicted MOS by (c) Baseline and (d) FB-IQA using Inception-Resnet-v2 [99] versus ground truth MOS. All models are trained and tested on KonIQ-10k [63] database which is divided to training set (7,058 images) and test set (2,015 images). Each point in the scatter plots corresponds to an image where x-axis and y-axis represent ground truth of the MOS and the predicted MOS, respectively.

RMSE, PLCC and SROCC, respectively. FB-IQA also performs well comparing Baseline when pretrained model is replaced with Inception-v4 [99]. In Fig. 6.4, the detail of distribution for each point corresponding to an image is shown by scatter plot. Compared with the Baseline, FB-IQA shows outperforming result and can improve learning accuracy of NR-IQA model with attaching three convolutional layers and filter concatenation between pretrained model and GAP layer. In this experiment, we have demonstrated the effectiveness of FB-IQA in application to NR-IQA and can be widely applied to various models.

6.5.4 Evaluation Performance

For fair comparison, conventional approaches including DIVIINE [73], CORNIA [112], BRISQUE [72], and HOSA [109] were trained and tested by the same strategy from original authors while deep learning approaches (DeepRN [104], Koncept [46], MetaIQA [122]) were trained and tested using same data transformation from proposed method. We use four different databases such as LIVE [89], CSIQ [55], LIVE-itW [34], and KonIQ-10k [63]. For each database, it is randomly divided into 70%, 20% and 10% of total images as training, testing and validating images without overlapping from each others. In Table 6.3, the best and second best results among them for each database are shown in bold-text on red and blue color, respectively. Based on this table, FB-IQA outperforms the existing methods in the best and the second best position by a large margin in PLCC and SROCC except Koncept [46] which performs the second best on LIVE database [89]. For conclusion, FB-IQA is superior comparing with state-of-the-art benchmark methods including conventional and deep learning

	PLCC ↑				SROCC ↑			
Methods	LIVE [89]	CSIQ [55]	LIVE-itW [34]	KonIQ-10k [63]	LIVE [89]	CSIQ [55]	LIVE-itW [34]	KonIQ-10k [63]
DIIVINE [73]	0.8434	0.6652	0.5652	0.6125	0.8233	0.6192	0.5462	0.5892
CORNIA [112]	0.9338	0.7783	0.6257	0.8086	0.9302	0.6961	0.5963	0.7893
BRISQUE [72]	0.9174	0.7906	0.6104	0.7074	0.9114	0.7013	0.5801	0.7058
HOSA [109]	0.9403	0.7893	0.6451	0.8281	0.9397	0.7083	0.6170	0.8057
DeepRN [104]	0.9777	0.9752	0.8181	0.9140	0.9776	0.9760	0.7411	0.8900
Koncept [46]	**0.9821**	0.9793	0.8030	0.9153	**0.9816**	0.9764	0.7334	0.8978
MetaIQA [122]	0.9766	0.9724	0.7743	0.8813	0.9753	0.9729	0.7193	0.8888
FB-IQA(Inception-v4 [99])(Ours)	0.9729	**0.9818**	**0.8318**	**0.9276**	0.9719	**0.9818**	**0.7598**	**0.9112**
FB-IQA(Inception-Resnet-v2 [99])(Ours)	**0.9833**	**0.9846**	**0.8237**	**0.9315**	**0.9828**	**0.9822**	**0.7692**	**0.9140**

Table 6.3 Comparison results (PLCC and SROCC) of FB-IQA with several state-of-the-art NR-IQA methods on artificial authentic distorted IQA databases (LIVE [89] and CSIQ [55]) and authentic distorted IQA databases (LIVE-itW [34] and KonIQ-10k [63]). For comparison, the best and second best results on PLCC and SROCC for each database are shown in bold-text on red and blue color, respectively.

approaches. With this quantitative evaluation, we can be concluded that FB-IQA performs well in four public databases.

6.5.5 Cross Database Evaluation

To evaluate the generalization capability, we trained FB-IQA on specific database and tested on entire databases. In Table 6.4, the value in the same training and testing database is produced from 70%, 20%, 10% of total images as trained, tested, and validated images without overlapping from each others. On the other hand, the value from different databases is trained by 70% of total images from trained database and tested by all images in tested database. Before we train in entire databases, we firstly normalize the subjective scores DMOS and MOS of images to be in the range started from 0 to 100 for fair comparison during cross evaluation. Cross database evaluation can be concluded that FB-IQA performs well in different subjective scores. In Table 6.5, the detail performance of proposed method and several NR-IQA method based on deep learning approach (DeepRN [104], Koncept [46], MetaIQA [122]) is trained on KonIQ-10k [63] database and tested on entire databases. Based on this table, FB-IQA is significantly improvement result comparing state-of-the-art benchmark methods by a large margin on all measures and the best results among them for each database and measure are shown boldfaced. We also report training on LIVE [89], CSIQ [55], and LIVE-itW[34] and testing on entire databases for individual NR-IQA method based on deep learning approach in supplementary material.

		Tested on			
		LIVE [89]	CSIQ [55]	LIVE-itW [34]	KonIQ-10k [63]
Trained on		MAE ↓			
	LIVE [89]	**3.3243**	13.4226	16.9102	20.9976
	CSIQ [55]	9.6095	**3.6838**	17.3657	22.8985
	LIVE-itW [34]	17.0191	17.1919	**8.7336**	10.2270
	KonIQ-10k [63]	15.4757	21.7364	13.1366	**4.3995**
		RMSE ↓			
	LIVE [89]	**4.2305**	17.3600	21.3135	26.6739
	CSIQ [55]	12.4470	**4.7841**	22.1622	28.5549
	LIVE-itW [34]	22.3248	20.4968	**11.6876**	13.2093
	KonIQ-10k [63]	17.9158	25.5198	15.3444	**5.7478**
		PLCC ↑			
	LIVE [89]	**0.9833**	0.7797	0.4183	0.5459
	CSIQ [55]	0.8715	**0.9846**	0.3673	0.5026
	LIVE-itW [34]	0.5151	0.6300	**0.8318**	0.7627
	KonIQ-10k [63]	0.8008	0.7183	0.8378	**0.9315**
		SROCC ↑			
	LIVE [89]	**0.9828**	0.7597	0.4000	0.5643
	CSIQ [55]	0.8830	**0.9822**	0.3610	0.5057
	LIVE-itW [34]	0.6207	0.5441	**0.7598**	0.6748
	KonIQ-10k [63]	0.8234	0.7130	0.8185	**0.9140**

Table 6.4 Cross database evaluation of FB-IQA (MAE, RMSE, PLCC, and SROCC) on five benchmark databases. FB-IQA is trained on specific database and tested on entire databases. For comparison, the best results on training set from the same database are shown in boldfaced.

Database	Method	MAE ↓	RMSE ↓	PLCC ↑	SROCC ↑
LIVE [89]	DeepRN [104]	16.7720	19.9585	0.6449	0.7087
	Koncept [46]	16.3148	18.9406	0.7266	0.7598
	meta-IQA [122]	16.4059	19.3515	0.6739	0.7124
	FB-IQA (Ours)	**15.4757**	**17.9158**	**0.8008**	**0.8234**
CSIQ [55]	DeepRN [104]	22.1963	25.7765	0.6476	0.6596
	Koncept [46]	22.1271	26.0032	0.6638	0.6534
	meta-IQA [122]	21.7430	25.7488	0.6640	0.6332
	FB-IQA (Ours)	**21.7364**	**25.5198**	**0.7183**	**0.7130**
LIVE-itW [34]	DeepRN [104]	13.3557	15.6656	0.7965	0.7834
	Koncept [46]	14.4944	16.7944	0.7913	0.7777
	meta-IQA [122]	13.5184	15.9939	0.7855	0.7752
	FB-IQA (Ours)	**13.1366**	**15.3444**	**0.8378**	**0.8185**
KONIQ-10k [63]	DeepRN [104]	5.4693	6.8236	0.9140	0.8900
	Koncept [46]	4.9327	6.4849	0.9153	0.8978
	meta-IQA [122]	5.2833	7.4815	0.8813	0.8888
	FB-IQA (Ours)	**4.3995**	**5.7478**	**0.9315**	**0.9140**

Table 6.5 Performance FB-IQA and several state-of-the-art NR-IQA methods with trained on KonIQ-10k [63] database and tested on entire databases. For comparison, the best results on all measures for each database are shown in boldfaced.

Database	Method	MAE ↓	RMSE ↓	PLCC ↑	SROCC ↑
LIVE [89]	DeepRN [104]	4.6829	5.8970	0.9777	0.9776
	Koncept [46]	3.9134	4.9609	0.9821	0.9816
	meta-IQA [122]	4.1084	5.1012	0.9766	0.9753
	FB-IQA (Ours)	**3.3243**	**4.2305**	**0.9833**	**0.9828**
CSIQ [55]	DeepRN [104]	14.8032	18.7419	0.7632	0.7307
	Koncept [46]	14.4880	18.4973	0.7644	0.7535
	meta-IQA [122]	15.2368	21.9710	0.6715	0.7586
	FB-IQA (Ours)	**13.4226**	**17.3600**	**0.7797**	**0.7597**
LIVE-itW [34]	DeepRN [104]	**16.8640**	**21.3090**	0.3686	0.3629
	Koncept [46]	17.0614	21.5023	0.3823	0.3811
	meta-IQA [122]	18.5168	25.9477	0.3076	0.3603
	FB-IQA (Ours)	16.9102	21.3135	**0.4183**	**0.4000**
KonIQ-10k [63]	DeepRN [104]	**18.1234**	**24.8225**	0.4631	0.5383
	Koncept [46]	21.9435	30.8545	0.4880	**0.5730**
	meta-IQA [122]	33.8798	61.7933	0.3603	0.5363
	FB-IQA (Ours)	20.9976	26.6739	**0.5459**	0.5643

Table 6.6 Performance FB-IQA and several state-of-the-art NR-IQA methods trained on LIVE [89] database and tested on entire databases. For comparison, the best results on all measures for each database are shown in boldfaced.

6.6 Additional Experimental Results

In this section, we present additional experimental results of FB-IQA and several state-of-the-art NR-IQA method based on deep learning approach such as DeepRN [104], Koncept [46] and MetaIQA [122]. Previously, strategy to separate training, testing and validating images has been explained on main paper. Table 6.6, 6.7, and 6.8 show the performance FB-IQA and state-of-the-art NR-IQA methods trained on LIVE [89], CSIQ [55] and LIVE-itW [34] databases and tested on entire databases, respectively.

In Table 6.6, FB-IQA performs the best when it is tested on LIVE [89] and CSIQ [55] databases for all measures and also shows good performance with small margin result from the best when it is tested on LIVE-itW [34] and KonIQ-10k [63] databases. Furthermore, table 6.7 shows that FB-IQA has obvious result compared to other methods by a large margin on all measures except PLCC and SROCC on LIVE [89] database and SROCC on KonIQ-10k

Database	Method	MAE ↓	RMSE ↓	PLCC ↑	SROCC ↑
LIVE [89]	DeepRN [104]	9.6555	13.0406	**0.8952**	0.8895
	Koncept [46]	10.1721	12.5469	0.8817	0.8794
	meta-IQA [122]	10.0386	12.6349	0.8900	**0.8938**
	FB-IQA (Ours)	**9.6095**	**12.4470**	0.8715	0.8830
CSIQ [55]	DeepRN [104]	5.3201	6.4792	0.9752	0.9760
	Koncept [46]	4.5013	5.6242	0.9793	0.9764
	meta-IQA [122]	5.2832	6.7181	0.9724	0.9729
	FB-IQA (Ours)	**3.6838**	**4.7841**	**0.9846**	**0.9822**
LIVE-itW [34]	DeepRN [104]	19.3726	24.4483	0.3212	0.2888
	Koncept [46]	18.5966	24.8297	0.3274	0.3362
	meta-IQA [122]	19.6462	24.9320	0.3639	0.3439
	FB-IQA (Ours)	**17.3657**	**22.1622**	**0.3673**	**0.3610**
KonIQ-10k [63]	DeepRN [104]	22.9241	29.7519	0.4966	0.4658
	Koncept [46]	25.6686	35.5859	0.4436	0.5223
	meta-IQA [122]	25.9595	33.8654	0.4826	**0.5282**
	FB-IQA (Ours)	**22.8985**	**28.5549**	**0.5026**	0.5057

Table 6.7 Performance FB-IQA and several state-of-the-art NR-IQA methods trained on CSIQ [55] database and tested on entire databases. For comparison, the best results on all measures for each database are shown in boldfaced.

Database	Method	MAE ↓	RMSE ↓	PLCC ↑	SROCC ↑
LIVE [89]	DeepRN [104]	16.2547	22.4086	0.5727	**0.6620**
	Koncept [46]	16.4702	22.8716	0.5578	0.6156
	meta-IQA [122]	**16.2296**	23.1367	**0.5734**	0.6604
	FB-IQA (Ours)	17.0191	**22.3248**	0.5151	0.6207
CSIQ [55]	DeepRN [104]	17.3626	20.5728	0.6236	0.5755
	Koncept [46]	17.4129	20.7814	0.6224	0.5578
	meta-IQA [122]	17.7591	21.1507	0.6203	**0.6158**
	FB-IQA (Ours)	**17.1919**	**20.4968**	**0.6300**	0.5441
LIVE-itW [34]	DeepRN [104]	9.1001	11.9741	0.8181	0.7411
	Koncept [46]	9.4824	12.4099	0.8030	0.7334
	meta-IQA [122]	10.1313	13.2536	0.7743	0.7193
	FB-IQA (Ours)	**8.7336**	**11.6876**	**0.8318**	**0.7598**
KonIQ-10k [63]	DeepRN [104]	10.6456	12.2106	0.7573	**0.7566**
	Koncept [46]	**10.3950**	**12.6297**	0.7349	0.7507
	meta-IQA [122]	10.6019	13.3978	0.7469	0.6806
	FB-IQA (Ours)	10.2270	13.2093	**0.7627**	0.6748

Table 6.8 Performance FB-IQA and several state-of-the-art NR-IQA methods trained on LIVE-itW [34] database and tested on entire databases. For comparison, the best results on all measures for each database are shown in boldfaced.

[63] database. When FB-IQA is trained on LIVE-itW [34] database, it also achieves the best on CSIQ [55] and LIVE-itW [34] database except SROCC measure on CSIQ [55] database in which Meta-IQA [122] perform the best presented on Table 6.8. Moreover, FB-IQA can be competed with other methods when it is tested on LIVE [89] and KonIQ-10k [63] databases with small margin for differences.

Based on our experiments, FB-IQA always achieves the best when it is trained and tested on the same database. It also performs well on different trained and tested databases comparing state-of-the-art NR-IQA methods for overall performance. Based on these tables, FB-IQA performs to be the best method comparing state-of-the-art NR-IQA methods to predict image quality by quantitative cross database evaluation.

6.7 Conclusions

In this chapter, we have demonstrated a new novel feature processing using different convolutional features. Unlike the previous methods which predict the MOS using DCNNs model, our proposed networks focus on separating features to collect information of object poses, image foreground position and blurry image background. Our architecture can exploit learning accuracy of NR-IQA based on Inception-Resnet-v2 as a base model and separating features using different convolutional kernel sizes on four public databases (LIVE, CSIQ, LIVE-itW, and KonIQ-10k). With same training and testing strategy, adding architecture of DCNNs with three convolutional boxes and one filter concatenation is one innovation of our approach in which it can burst learning accuracy comparing Baseline model. In our experiments, we have clearly shown that FB-IQA architecture can improve the performance of learning accuracy with respect to previous state-of-the-art benchmark methods. We have used four public databases and showed that FB-IQA is an effective way to increase learning accuracy and it can be widely applied to various models. We expect the next steps of our work to be taken towards a more attractive for feature extraction of DCNNs, and focus on integrating DCNNs architecture to solve artificially and authentically distorted IQA databases.

Chapter 7

Conclusion

This thesis has investigated a new topic for controlling fish feeding machine using feature extraction of nutriment and ripple behavior. This thesis has made a number of significant contributions to the aquaculture field. In addition, we provide suggestions for areas that warrant future attention.

The organization of this chapter is as follows. Section 7.1 presents conclusion and discussion of the thesis. We then explain limitation and future work of proposed method in section 7.2 and 7.3.

7.1 Conclusion and Discussion

Feature extraction in application to aquaculture is important feature to optimize the amount of nutriment giving to the fish. Feature extraction of nutriment and ripple behavior has presented an agreement to complement each other. Recent studies have shown that it is possible to track all detected objects in sequence frames on video. However, there is no agreement to track multiple small and dense nutriments and also to detect ripple activity area in the video. Feature extraction of nutriment and ripple behavior has been presented and demonstrated to be promising for optimizing fish feeding process to give an optimal rate of both costs and profits. We have demonstrated an area with and without ripple activity which has a big difference gap and the method consistently performs well on the small and dense nutriments videos. Image quality assessment can also be suggested method for ripple behavior to calculate activity of ripple. We also show the result using image quality assessment architecture to predict the activity of ripple with and without training and testing dataset.

7.2 Limitation

There are several limitations of this thesis which is described in below:

- Large dataset is an issue of this research because collecting the video has limited location in Kyushu area.
- To prove that IQA model can apply to fish feeding application, we need to manually annotating the video of fish feeding and train IQA model to that dataset.
- Dataset annotation should be created for easier evaluation process.
- Video which has different weather conditions should be added to the new dataset to make robust result comparing with specific weather.

7.3 Future Work

We expect proposed method to open the door for future work and to go beyond for developing a large community-generated database and focus on integrating with the sensors to give more accurate and robust results. We also expect to build whole automatic system to apply in other fish feeding datasets and create the system to estimate degree of hunger of fishes.

7.4 Contribution on Computer Vision Community

FB-IQA creates several contributions for increasing learning accuracy using adding some convolutional features. This method can burst performance of CNN model with same parameter at training stage.

References

[FAO] FAO Fisheries & Aquaculture Global Aquaculture Production, Overview.

[Ngh] Simple video stabilization using OpenCV | Nghia Ho.

[3] Adegboye, M. A., Aibinu, A. M., Kolo, J. G., Aliyu, I., Folorunso, T. A., and Lee, S. (2020). Incorporating intelligence in fish feeding system for dispensing feed based on fish feeding intensity. *IEEE Access*, 8:91948–91960.

[4] Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., and Gool, L. V. (2019). Generative adversarial networks for extreme learned image compression.

[5] Aliyu, I., Gana, K., Musa, A., Agajo, J., Orire, A., Abiodun, F., and Adegboye, M. (2017). A proposed fish counting algorithm using digital image processing technique. *ATBU Journal of Science, Technology and Education*, 5(1):1–11.

[6] Andrew, J. E., Noble, C., Kadri, S., Jewell, H., and Huntingford, F. A. (2002). The effect of demand feeding on swimming speed and feeding responses in atlantic salmon salmo salar l., gilthead sea bream sparus aurata l. and european sea bass dicentrarchus labrax l. in sea cages. *Aquaculture Research*, 33(7):501–507.

[7] Asaeda, T., Vu, T. K., and Manatunge, J. (2005a). Effects of flow velocity on feeding behavior and microhabitat selection of the stone moroko pseudorasbora parva: A trade-off between feeding and swimming costs. *Transactions of the American Fisheries Society*, 134(2):537–547.

[8] Asaeda, T., Vu, T. K., and Manatunge, J. (2005b). Effects of flow velocity on feeding behavior and microhabitat selection of the stone moroko pseudorasbora parva: A trade-off between feeding and swimming costs. *Transactions of the American Fisheries Society*, 134(2):537–547.

[9] Atoum, Y., Srivastava, S., and Liu, X. (2015a). Automatic feeding control for dense aquaculture fish tanks. *IEEE Signal Processing Letters*, 22(8):1089–1093.

[10] Atoum, Y., Srivastava, S., and Liu, X. (2015b). Automatic feeding control for dense aquaculture fish tanks. *IEEE Signal Processing Letters*, 22(8):1089–1093.

[11] Auliar, R. and Bekaroo, G. (2020). A smart fish feeding system for internet of things based aquariums. In *Proceedings of the 2nd International Conference on Intelligent and Innovative Computing Applications*, ICONIC '20, New York, NY, USA. Association for Computing Machinery.

References

[12] Babaee, M., Li, Z., and Rigoll, G. (2019). A dual cnn–rnn for multiple people tracking. *Neurocomputing*, 368:69–83.

[13] Barron, J. M., Hanson, K. C., Headley, R. R., Hawke, K. A., Twibell, R. G., and Gannam, A. L. (2020). Evaluation of effluent waste water from salmonid culture as a potential food and water supply for culturing larval pacific lamprey entosphenus tridentatus. *Aquaculture*, 517:734791.

[14] Berclaz, J., Fleuret, F., and Fua, P. (2006). Robust people tracking with global trajectory optimization. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 744–750.

[15] Berclaz, J., Fleuret, F., Turetken, E., and Fua, P. (2011). Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819.

[16] Bewley, A., Ge, Z., Ott, L., Ramos, F., and Upcroft, B. (2016). Simple online and realtime tracking. *2016 IEEE International Conference on Image Processing (ICIP)*.

[17] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020a). Yolov4: Optimal speed and accuracy of object detection.

[18] Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020b). Yolov4: Optimal speed and accuracy of object detection.

[19] Breitenstein, M. D., Reichlin, F., Leibe, B., Koller-Meier, E., and Van Gool, L. (2009). Robust tracking-by-detection using a detector confidence particle filter. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1515–1522.

[20] Bridger, C. and Booth, R. (2003). The effects of biotelemetry transmitter presence and attachment procedures on fish physiology and behavior. *Reviews in Fisheries Science*, 11(1):13–34.

[21] Choi, Y., El-Khamy, M., and Lee, J. (2019). Variable rate deep image compression with a conditional autoencoder.

[22] Conti, S. G., Roux, P., Fauvel, C., Maurer, B. D., and Demer, D. A. (2006). Acoustical monitoring of fish density, behavior, and growth rate in a tank. *Aquaculture*, 251(2):314–323.

[23] Costa, C., Loy, A., Cataudella, S., Davis, D., and Scardi, M. (2006). Extracting fish size using dual underwater cameras. *Aquacultural Engineering*, 35(3):218–227.

[24] de los Santos, C. B., Olivé, I., Moreira, M., Silva, A., Freitas, C., Araújo Luna, R., Quental-Ferreira, H., Martins, M., Costa, M. M., Silva, J., Cunha, M. E., Soares, F., Pousão-Ferreira, P., and Santos, R. (2020). Seagrass meadows improve inflowing water quality in aquaculture ponds. *Aquaculture*, 528:735502.

[25] Deng, J., Dong, W., Socher, R., Li, L., Kai Li, and Li Fei-Fei (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

[26] Ding, G., Chen, W., Zhao, S., Han, J., and Liu, Q. (2018). Real-time scalable visual tracking via quadrangle kernelized correlation filters. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):140–150.

[27] Ding, G., Guo, Y., Chen, K., Chu, C., Han, J., and Dai, Q. (2019). Decode: Deep confidence network for robust image classification. *IEEE Transactions on Image Processing*, 28(8):3752–3765.

[28] Duarte, S., Reig, L., and Oca, J. (2009). Measurement of sole activity by digital image analysis. *Aquacultural Engineering*, 41(1):22–27.

[29] Ess, A., Leibe, B., Schindler, K., and Van Gool, L. (2008). A mobile vision system for robust multi-person tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[30] Ess, A., Leibe, B., and Van Gool, L. (2007). Depth and appearance for mobile scene analysis. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8.

[31] Farheen, U., Preeti, H. M., and Gadgay, B. (2018). Automatic controlling of fish feeding system. *International Journal for Research in Applied Science and Engineering Technology*, 6(7):362–366.

[32] Fazio, F. (2019). Fish hematology analysis as an important tool of aquaculture: A review. *Aquaculture*, 500:237–242.

[33] Freitas, J., Vaz-Pires, P., and Câmara, J. S. (2020). From aquaculture production to consumption: Freshness, safety, traceability and authentication, the four pillars of quality. *Aquaculture*, 518:734857.

[34] Ghadiyaram, D. and Bovik, A. C. (2016). Massive online crowdsourced study of subjective and objective picture quality. *IEEE Transactions on Image Processing*, 25(1):372–387.

[35] Ghadiyaram, D. and Bovik, A. C. (2016). Perceptual quality prediction on authentically distorted images using a bag of features approach. *CoRR*, abs/1609.04757.

[36] Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., and Cord, M. (2020). Learning representations by predicting bags of visual words.

[37] Golestaneh, S. and Karam, L. J. (2016). Reduced-reference quality assessment based on the entropy of dwt coefficients of locally weighted gradient magnitudes. *Trans. Img. Proc.*, 25(11):5293–5303.

[38] Gu, K., Zhai, G., Yang, X., and Zhang, W. (2015). Using free energy principle for blind image quality assessment. *IEEE Transactions on Multimedia*, 17(1):50–63.

[39] Guillen, J., Natale, F., Carvalho, N., Casey, J., Hofherr, J., Druon, J.-N., Fiore, G., Gibin, M., Zanzi, A., and Martinsohn, J. T. (2018). Global seafood consumption footprint. *Ambio*, 48(2):111–122.

[40] Hashimoto, Y., Hama, H., and Zin, T. (2020). Robust tracking of cattle using super pixels and local graph cut for monitoring systems. *International Journal of Innovative Computing, Information and Control (IJICIC)*, 16(4).

[41] HASSAN, S. G., HASAN, M., and LI, D. (2016). Information fusion in aquaculture: a state-of the art review. *Frontiers of Agricultural Science and Engineering*, 3(3):206.

[42] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

[43] Higuchi, K., Yoshida, K., Gen, K., Matsunari, H., Takashi, T., Mushiake, K., and Soyano, K. (2018). Effect of long-term food restriction on reproductive performances in female yellowtail, seriola quinqueradiata. *Aquaculture*, 486:224–231.

[44] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors.

[45] Horie, J., Mitamura, H., Ina, Y., Mashino, Y., Noda, T., Moriya, K., Arai, N., and Sasakura, T. (2017). Development of a method for classifying and transmitting high-resolution feeding behavior of fish using an acceleration pinger. *Animal Biotelemetry*, 5(1).

[46] Hosu, V., Lin, H., Sziranyi, T., and Saupe, D. (2020). Koniq-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Transactions on Image Processing*, 29:4041–4056.

[47] Hung, C.-C., Tsao, S.-C., Huang, K.-H., Jang, J.-P., Chang, H.-K., and Dobbs, F. C. (2016). A highly sensitive underwater video system for use in turbid aquaculture ponds. *Scientific Reports*, 6(1).

[48] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.

[49] J. Guillen, F. Natale, N. C. (2019). Global seafood consumption footprint. *Ambio*, 48:111–122.

[50] Jiang, H., Fels, S., and Little, J. J. (2007). A linear programming approach for multiple object tracking. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[51] Kim, J. and Lee, S. (2017). Deep learning of human visual sensitivity in image quality assessment framework. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1969–1977.

[52] Kim, J., Zeng, H., Ghadiyaram, D., Lee, S., Zhang, L., and Bovik, A. C. (2017). Deep convolutional neural models for picture-quality prediction: Challenges and solutions to data-driven image quality assessment. *IEEE Signal Processing Magazine*, 34(6):130–141.

[53] Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization.

[54] Krizhevsky, A., Sutskever, I., and Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25.

[55] Larson, E. C. and Chandler, D. M. (2010). Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1):1–21.

[56] Leal-Taixé, L., Fenzi, M., Kuznetsova, A., Rosenhahn, B., and Savarese, S. (2014). Learning an image-based motion context for multiple people tracking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3542–3549.

[57] Leal-Taixé, L., Milan, A., Reid, I., Roth, S., and Schindler, K. (2015). Motchallenge 2015: Towards a benchmark for multi-target tracking.

[58] Leal-Taixé, L., Pons-Moll, G., and Rosenhahn, B. (2011). Everybody needs somebody: Modeling social and grouping behavior on a linear programming multiple people tracker. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 120–127.

[59] Li, B., Wu, W., Wang, Q., Zhang, F., Xing, J., and Yan, J. (2018). Siamrpn++: Evolution of siamese visual tracking with very deep networks.

[60] Li, L., Lin, W., Wang, X., Yang, G., Bahrami, K., and Kot, A. C. (2016). No-reference image blur assessment based on discrete orthogonal moments. *IEEE Transactions on Cybernetics*, 46(1):39–50.

[61] Li, L., Zhu, H., Yang, G., and Qian, J. (2014). Referenceless measure of blocking artifacts by tchebichef kernel analysis. *IEEE Signal Processing Letters*, 21(1):122–125.

[62] Li Zhang, Yuan Li, and Nevatia, R. (2008). Global data association for multi-object tracking using network flows. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8.

[63] Lin, H., Hosu, V., and Saupe, D. (2018). Koniq-10k: Towards an ecologically valid and large-scale IQA database. *CoRR*, abs/1803.08489.

[64] Lin, W. (2017). Real-time multi-object tracking with hyper-plane matching.

[65] Liu, H., Li, H., Wei, H., Zhu, X., Han, D., Jin, J., Yang, Y., and Xie, S. (2019a). Biofloc formation improves water quality and fish yield in a freshwater pond aquaculture system. *Aquaculture*, 506:256–269.

[66] Liu, J., Wang, S., and Urtasun, R. (2019b). Dsic: Deep stereo image compression.

[67] Liu, W., Ke, H., Xie, J., Tan, H., Luo, G., Xu, B., and Abakari, G. (2020). Characterizing the water quality and microbial communities in different zones of a recirculating aquaculture system using biofloc biofilters. *Aquaculture*, 529:735624.

[68] Liu, X., Suganuma, M., Sun, Z., and Okatani, T. (2019c). Dual residual networks leveraging the potential of paired operations for image restoration.

[69] Liu, Z., Li, X., Fan, L., Lu, H., Liu, L., and Liu, Y. (2014). Measuring feeding activity of fish in ras using computer vision. *Aquacultural Engineering*, 60:20–27.

[70] Mastan, I. D. and Raman, S. (2019). Multi-level encoder-decoder architectures for image restoration.

[71] Milan, A., Leal-Taixe, L., Reid, I., Roth, S., and Schindler, K. (2016). Mot16: A benchmark for multi-object tracking.

[72] Mittal, A., Moorthy, A. K., and Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708.

[73] Moorthy, A. K. and Bovik, A. C. (2011). Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE Transactions on Image Processing*, 20(12):3350–3364.

[74] Murray, S. (2017). Real-time multiple object tracking - a study on the importance of speed.

[75] Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814. Omnipress.

[76] Noda, T., Kawabata, Y., Arai, N., Mitamura, H., and Watanabe, S. (2014). Animal-mounted gyroscope/accelerometer/magnetometer: In situ measurement of the movement performance of fast-start behaviour in fish. *Journal of Experimental Marine Biology and Ecology*, 451:55–68.

[77] Oostlander, P., van Houcke, J., Wijffels, R., and Barbosa, M. (2020). Microalgae production cost in aquaculture hatcheries. *Aquaculture*, 525:735310.

[78] Pang, B., Li, Y., Zhang, Y., Li, M., and Lu, C. (2020). Tubetk: Adopting tubes to track multi-object in a one-step training model.

[79] Pautsina, A., Císař, P., Štys, D., Terjesen, B. F., and Åsa Maria O. Espmark (2015). Infrared reflection system for indoor 3d tracking of fish. *Aquacultural Engineering*, 69:7–17.

[80] Pellegrini, S., Ess, A., Schindler, K., and van Gool, L. (2009). You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th International Conference on Computer Vision*, pages 261–268.

[81] Pirsiavash, H., Ramanan, D., and Fowlkes, C. C. (2011). Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*, pages 1201–1208.

[82] Pradana, H. and Horio, K. (2020). Tuna nutriment tracking using trajectory mapping in application to aquaculture fish tank. In *2020 Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8.

[83] Qiang, G., XinTing, Y., Chao, Z., Kai, L., ChuanHeng, S., and Ming, C. (2018). Fish feeding behavior detection method based on shape and texture features. *Journal of Shanghai Ocean University*, 27(2):181–189.

[84] Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement.

[85] Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster r-cnn: Towards real-time object detection with region proposal networks.

[86] Rillahan, C., Chambers, M., Howell, W. H., and Watson, W. H. (2009). A self-contained system for observing and quantifying the behavior of atlantic cod, gadus morhua, in an offshore aquaculture cage. *Aquaculture*, 293(1):49–56.

[87] Saad, M. A., Bovik, A. C., and Charrier, C. (2012). Blind image quality assessment: A natural scene statistics approach in the dct domain. *IEEE Transactions on Image Processing*, 21(8):3339–3352.

[88] Saberioon, M., Gholizadeh, A., Cisar, P., Pautsina, A., and Urban, J. (2017). Application of machine vision systems in aquaculture with emphasis on fish: state-of-the-art and key issues. *Reviews in Aquaculture*, 9(4):369–387.

[89] Sheikh, H. R., Sabir, M. F., and Bovik, A. C. (2006). A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 15(11):3440–3451.

[90] Simonyan, K. and Zisserman, A. (2015a). Very deep convolutional networks for large-scale image recognition.

[91] Simonyan, K. and Zisserman, A. (2015b). Very deep convolutional networks for large-scale image recognition.

[92] Soetedjo, A. and Somawirata, I. (2019). Improving traffic sign detection by combining mser and lucas kanade tracking. *International Journal of Innovative Computing, Information and Control (IJICIC)*, 15(2).

[93] Soto-Zarazúa, G. M., Rico-García, E., Ocampo, R., Guevara-González, R. G., and Herrera-Ruiz, G. (2009). Fuzzy-logic-based feeder system for intensive tilapia production (oreochromis niloticus). *Aquaculture International*, 18(3):379–391.

[94] Srivastava, A., Lee, A. B., Simoncelli, E. P., and Zhu, S. C. (2002). On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*.

[95] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.

[96] Stien, L. H., Bratland, S., Austevoll, I., Oppedal, F., and Kristiansen, T. S. (2007). A video analysis procedure for assessing vertical fish distribution in aquaculture tanks. *Aquacultural Engineering*, 37(2):115–124.

[97] Stoner, A. W., Ottmar, M. L., and Hurst, T. P. (2006). Temperature affects activity and feeding motivation in pacific halibut: Implications for bait-dependent fishing. *Fisheries Research*, 81(2):202–209.

[98] Subasinghe, R., Soto, D., and Jia, J. (2009). Global aquaculture and its role in sustainable development. *Reviews in Aquaculture*, 1(1):2–9.

[99] Szegedy, C., Ioffe, S., and Vanhoucke, V. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261.

[100] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going deeper with convolutions.

[101] Tang, S., Andriluka, M., Andres, B., and Schiele, B. (2017). Multiple people tracking by lifted multicut and person re-identification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3701–3710.

[102] Tanoue, H., Komatsu, T., Tsujino, T., Suzuki, I., Watanabe, M., Goto, H., and Miyazaki, N. (2012). Feeding events of japanese lates lates japonicus detected by a high-speed video camera and three-axis micro-acceleration data-logger. *Fisheries Science*, 78(3):533–538.

[103] Ueno, T., Gu, Q., Aoyama, T., Takaki, T., Ishii, I., and Kawahara, T. (2015). Motion-blur-free microscopic video shooting based on frame-by-frame intermittent tracking. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 837–842.

[104] Varga, D., Saupe, D., and Szirányi, T. (2018). Deeprn: A content preserving deep architecture for blind image quality assessment. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

[105] Wang, Z., Zheng, L., Liu, Y., Li, Y., and Wang, S. (2020). Towards real-time multi-object tracking.

[Weisstein] Weisstein, E. W. Least Squares Fitting–Polynomial.

[107] Wu, T.-H., Huang, Y.-I., and Chen, J.-M. (2015). Development of an adaptive neural-based fuzzy inference system for feeding decision-making assessment in silver perch (bidyanus bidyanus) culture. *Aquacultural Engineering*, 66:41–51.

[108] Xu, J., Liu, Y., Cui, S., and Miao, X. (2006). Behavioral responses of tilapia (oreochromis niloticus) to acute fluctuations in dissolved oxygen levels as monitored by computer vision. *Aquacultural Engineering*, 35(3):207–217.

[109] Xu, J., Ye, P., Li, Q., Du, H., Liu, Y., and Doermann, D. (2016). Blind image quality assessment based on high order statistics aggregation. *IEEE Transactions on Image Processing*, 25(9):4444–4457.

[110] Xue, W., Mou, X., Zhang, L., Bovik, A. C., and Feng, X. (2014). Blind image quality assessment using joint statistics of gradient magnitude and laplacian features. *IEEE Transactions on Image Processing*, 23(11):4850–4862.

[111] Xue, W., Zhang, L., and Mou, X. (2013). Learning without human scores for blind image quality assessment. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 995–1002.

[112] Ye, P., Kumar, J., Kang, L., and Doermann, D. (2012). Unsupervised feature learning framework for no-reference image quality assessment. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1098–1105.

[113] Yu, F., Li, W., Li, Q., Liu, Y., Shi, X., and Yan, J. (2016). Poi: Multiple object tracking with high performance detection and appearance feature.

[114] Zamir, A. R., Dehghan, A., and Shah, M. (2012). GMCP-tracker: Global multi-object tracking using generalized minimum clique graphs. In *Computer Vision – ECCV 2012*, pages 343–356. Springer Berlin Heidelberg.

[115] Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M. H., and Shao, L. (2020). Cycleisp: Real image restoration via improved data synthesis.

[116] Zhang, X. and Izquierdo, E. (2019). Real-time multi-target multi-camera tracking with spatial-temporal information. In *2019 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4.

[117] Zhang, Y., Wang, C., Wang, X., Zeng, W., and Liu, W. (2020). Fairmot: On the fairness of detection and re-identification in multiple object tracking.

[118] Zhao, S., Ding, W., Zhao, S., and Gu, J. (2019). Adaptive neural fuzzy inference system for feeding decision-making of grass carp (ctenopharyngodon idellus) in outdoor intensive culturing ponds. *Aquaculture*, 498:28–36.

[119] Zhou, C., Lin, K., Xu, D., Chen, L., Guo, Q., Sun, C., and Yang, X. (2018). Near infrared computer vision and neuro-fuzzy model-based feeding decision system for fish in aquaculture. *Computers and Electronics in Agriculture*, 146:114–124.

[120] Zhou, C., Xu, D., Chen, L., Zhang, S., Sun, C., Yang, X., and Wang, Y. (2019). Evaluation of fish feeding intensity in aquaculture using a convolutional neural network and machine vision. *Aquaculture*, 507:457–465.

[121] Zhou, C., Zhang, B., Lin, K., Xu, D., Chen, C., Yang, X., and Sun, C. (2017). Near-infrared imaging to quantify the feeding behavior of fish in aquaculture. *Computers and Electronics in Agriculture*, 135:233–241.

[122] Zhu, H., Li, L., Wu, J., Dong, W., and Shi, G. (2020). Metaiqa: Deep meta-learning for no-reference image quality assessment.

[123] Zhu, Z., Wang, Q., Li, B., Wu, W., Yan, J., and Hu, W. (2018). Distractor-aware siamese networks for visual object tracking.

[124] Zion, B. (2012). The use of computer vision technologies in aquaculture – a review. *Computers and Electronics in Agriculture*, 88:125–132.

[125] Zion, B., Alchanatis, V., Ostrovsky, V., Barki, A., and Karplus, I. (2007). Real-time underwater sorting of edible fish species. *Computers and Electronics in Agriculture*, 56(1):34–45.